

# GMT 模块手册

GMT 中文社区

2016 年 06 月 13 日

1	backtracker	12
2	blockmean	13
3	blockmedian	14
4	blockmode	15
5	dimfilter	16
6	filter1d	17
7	fitcircle	19
8	gmt	22
9	gmt2kml	23
10	gmt5syntax	24
11	gmtconnect	25
12	gmtconvert	27
13	gmtdefaults	28
14	gmtflexure	29
15	gmtget	30
16	gmtgravmag3d	31
17	gmtinfo	32

18	gmtlogo	34
19	gmtmath	35
20	gmtregress	36
21	gmtselect	37
22	gmtset	40
23	gmtsimplify	42
24	gmtspatial	43
25	gmtswitch	44
26	gmtvector	45
27	gmtwhich	46
28	gravfft	47
29	grd2cpt	48
30	grd2rgb	49
31	grd2xyz	51
32	grdblend	53
33	grdclip	55
34	grdcontour	57
35	grdcut	58
36	grdedit	60
37	grdfft	62
38	grdfilter	63
39	grdflexure	64
40	grdgradient	65
41	grdgravmag3d	66
42	grdhisteq	67
43	grdimage	68

44	grdinfo	70
45	grdlandmask	73
46	grdmask	75
47	grdmath	77
48	grdpaste	78
49	grdpmodeler	79
50	grdproject	80
51	grdraSTER	82
52	grdredpol	84
53	grdconvert	85
54	grdrotater	87
55	grdsample	88
56	grdseamount	90
57	grdspotter	91
58	grdtrack	92
59	grdtrend	93
60	grdvector	95
61	grdview	97
62	grdvolume	98
63	greenspline	100
64	gshhg	101
65	hotspotter	102
66	img2google	103
67	img2grd	104
68	isogmt	105
69	kml2gmt	107

70	makecpt	109
71	mapproject	110
72	mgd77convert	112
73	mgd77info	113
74	mgd77list	114
75	mgd77magref	115
76	mgd77manage	116
77	mgd77path	117
78	mgd77sniffer	118
79	mgd77track	119
80	nearneighbor	120
81	originator	121
82	project	122
83	psbasemap	124
84	psclip	133
85	pscoast	135
86	pscontour	141
87	psconvert	142
88	pscoupe	147
89	pshistogram	154
90	psimage	156
91	pslegend	158
92	psmask	166
93	psmeca	167
94	pspolar	175
95	psrose	180

96	psscale	182
97	pssegy	184
98	pssegyz	185
99	pstext	186
100	psvelo	192
101	pswiggle	198
102	psxy	199
103	psxyz	218
104	rotconverter	219
105	sample1d	220
106	segy2grd	222
107	spectrum1d	223
108	sph2grd	224
109	sphdistance	225
110	sphinterpolate	226
111	sphtriangulate	227
112	splitxyz	228
113	surface	229
114	talwani2d	230
115	talwani3d	231
116	trend1d	232
117	trend2d	233
118	triangulate	234
119	2sys_binlist	235
120	2sys_cross	236
121	2sys_datalist	237

<a href="#">122x2sys__get</a>	238
<a href="#">123x2sys__init</a>	239
<a href="#">124x2sys__list</a>	240
<a href="#">125x2sys__merge</a>	241
<a href="#">126x2sys__put</a>	242
<a href="#">127x2sys__report</a>	243
<a href="#">128x2sys__solve</a>	244
<a href="#">129xyz2grd</a>	245

欢迎来到 GMT (Generic Mapping Tools) 的世界。

本项目是 GMT 中文社区维护的 GMT 模块手册。据不完全统计，GMT 中约有 130 个模块，本文档中将介绍其中常用的模块。

## 相关链接

- 社区主页：<http://gmt-china.org>
- 项目主页：<http://modules.gmt-china.org>
- 项目源码：[https://github.com/gmt-china/GMT\\_modules](https://github.com/gmt-china/GMT_modules)
- GMT 中文参考手册：<http://docs.gmt-china.org>

## 文档约定

所有的模块，在介绍其用法时，都尽量遵循如下约定：

- 不介绍 GMT 通用选项，只介绍模块专有的选项
- 中括号 [ ] 括起来的项表明该项是可选项
- 尖括号 < > 括起来的项表明实际使用时需要用具体的数值替代
- 用 | 分隔的项表明各项之间是可选的
- 用 ... 表示省略，即当前命令仅用于演示某些功能，命令本身是不完整的，无法直接执行
- # 之后的内容为注释

比如，`-A[a|c|d]<scale>[/<dist>]`：

- `<scale>` 是必须的，使用时需要用具体数字代替
- `[a|c|d]` 是可选的，实际使用时只能使用 `a` 或 `c` 或 `d`
- `[/<dist>]` 是可选的，使用时需要用具体数字代替

## 按功能分类

下面将 GMT 中的模块按照功能分类，并用一句话简述其功能。

- 主程序/脚本
  - *gmt* : GMT 主程序
  - *isogmt* : 在“隔离模式”下运行 GMT 命令或脚本
  - *gmt5syntax* : 将旧的 GMT 脚本修改成新的 *gmt <module>* 语法
- 绘制 1D 和 2D 数据
  - *psbasemap* : 绘制底图
  - *pscoast* : 在地图上绘制海岸线、河流、国界线
  - *psxy* : 在图上绘制线段、多边形和符号
  - *pstext* : 在图上写文本
  - *psscale* : 在图上绘制灰色或彩色色标
  - *psclip* : 打开或关闭多边形裁剪路径
  - *psimage* : 将图片或 EPS 文件放在地图上
  - *pslegend* : 绘制图例
  - *pshistogram* : 统计并绘制直方图
  - *psrose* : 绘制极坐标下的直方图 (sector 图、rose 图或 windrose 图)
  - *psmeca* : 在地图上绘制震源机制解
  - *pspolar* : 在震源球上绘制台站极性
  - *psvelo* : 在地图上绘制速度矢量、十字线、楔形图
  - *pscoupe* : 绘制震源机制解的剖面图
  - *grdvector* : 根据两个网格文件绘制矢量场
  - *grdimage* : 在图上绘制网格数据
  - *gmtlogo* : 在图上绘制 GMT 图形 logo
- 信息提取
  - *gmtget* : 列出单个或多个 GMT 参数的当前值
  - *gmtset* : 修改单个或多个 GMT 参数的值
  - *gmtdefaults* : 列出所有 GMT 参数的当前值
  - *gmtinfo* : 从表数据中提取信息
  - *grdinfo* : 从网格文件中提取基本信息



- *gmtselect* : 根据多个准则筛选数据
  - *gmtwhich* : 返回指定文件的完整路径
  - *fitcircle* : 拟合球面上数据点的平均位置及圆弧
- 1D 数据处理
  - *filter1d* : 对 1D 表数据做时间域滤波
  - *gmtsimplify* : 使用 Douglas-Peucker 算法对线段做简化
  - *gmtconnect* : 将端点接近的线段连接起来
  - *sample1d* : 对 1D 表数据进行重采样
  - *project* : 将数据点投影到线或大圆路径上, 生成测线, 坐标转换
- 2D 数据处理
  - *grdedit* : 修改网格文件的头段或内容
  - *grdcut* : 从一个网格文件中裁剪出一个子区域
  - *grdblend* : 将多个部分重叠的网格文件合并成一个网格文件
  - *grdpaste* : 将两个网格沿着其共同边界拼接成一个文件
  - *grdraster* : 从二进制数据中提取子区域并保存为 GMT 网格文件
  - *grdclip* : 对网格文件的 Z 值做裁剪
  - *grdlandmask* : 根据海岸线数据创建陆地-海洋的 mask 网格文件
  - *grdtrend* : 拟合网格的趋势面并计算残差
  - *grdsample* : 对网格文件做重采样
  - *grdvolume* : 计算网格数据中某个等值线所包围的表面积和体积
  - *grdproject* : 对网格数据做地图变换和逆变换
  - *grdmask* : 根据多边形数据或点数据创建 mask 网格文件
  - *grdconvert* : 在不同的网格格式之间互相转换
- 格式转换
  - *kml2gmt* : 将 Google Earth 的 KML 文件转换为 GMT 表数据
  - *psconvert* : 将 GMT 生成的 PS 文件转换为其他图片格式
  - *grd2xyz* : 将网格文件转换成表数据
  - *xyz2grd* : 将 XYZ 数据或 Z 数据转换成网格文件
  - *grd2rgb* : 将网格文件、原始 RGB 文件、SUN 光栅文件转换成 R/G/B 三个网格文件
- 计划整理的命令

1. *grdtrack*
  2. *makecpt*
  3. *grd2cpt*
  4. *gmtconvert*
  5. *trend1d*
  6. *grdcontour*
  7. *psmask*
  8. *grdview*
  9. *grdgradient*
  10. *grdmath*
  11. *gmt2kml*
  12. *trend2d*
  13. *mapproject*
  14. *surface*
  15. *grdfilter*
  16. *grdfft*
  17. *splitxyz*
- 没有整理计划的命令
    1. *blockmean*
    2. *blockmedian*
    3. *blockmode*
    4. *psxyz*
    5. *pscontour*
    6. *pssegy*
    7. *pssegyz*
    8. *pswiggle*
    9. *grdgravmag3d*
    10. *grdhisteq*
    11. *grdpmodeler*
    12. *grdredpol*

13. *grdrotater*
14. *dimfilter*
15. *triangulate*
16. *nearneighbor*
17. *mgd77convert*
18. *mgd77info*
19. *mgd77list*
20. *mgd77magref*
21. *mgd77manage*
22. *mgd77path*
23. *mgd77sniffer*
24. *mgd77track*
25. *x2sys\_binlist*
26. *x2sys\_cross*
27. *x2sys\_datalist*
28. *x2sys\_get*
29. *x2sys\_init*
30. *x2sys\_list*
31. *x2sys\_merge*
32. *x2sys\_put*
33. *x2sys\_report*
34. *x2sys\_solve*
35. *gmtswitch*
36. *backtracker*
37. *gmtgravmag3d*
38. *gmtspatial*
39. *gmtvector*
40. *gravfft*
41. *grdseamount*
42. *grdspotter*

- 43. [\*greenspline\*](#)
- 44. [\*gshhg\*](#)
- 45. [\*hotspotter\*](#)
- 46. [\*img2google\*](#)
- 47. [\*img2grd\*](#)
- 48. [\*originator\*](#)
- 49. [\*rotconverter\*](#)
- 50. [\*segy2grd\*](#)
- 51. [\*spectrum1d\*](#)
- 52. [\*sph2grd\*](#)
- 53. [\*sphdistance\*](#)
- 54. [\*sphinterpolate\*](#)
- 55. [\*sphtriangulate\*](#)

---

**backtracker**

---

官方文档 [backtracker](#)

简介

---

**blockmean**

---

官方文档 [blockmean](#)

简介

---

**blockmedian**

---

官方文档 [blockmedian](#)

简介

---

**blockmode**

---

官方文档 [blockmode](#)

简介



---

**dimfilter**

---

官方文档 [dimfilter](#)

---

filter1d

---

官方文档 [filter1d](#)

简介 对 1D 表数据做时间域滤波

## 选项

**-D<increment>** 当输入的时间序列是不等间隔采样时，需要使用该选项。<increment> 为输出数据的分辨率，所有横坐标（时间）都会被 rounded off 到 <increment> 的整数倍。当然，也可以使用 [sample1d](#) 对时间序列做重采样。

**-E** 输出时间序列的首尾端数据。默认情况下，首尾两端都会丢失半滤波器宽度的数据点

**-F<type><width>[<mode>]** 设置滤波器类型

滤波器分为两大类，卷积滤波器和非卷积滤波器。<type> 用于指定滤波器类型，<width> 指定滤波器宽度（单位与时间数据相同）。

对于卷积滤波器，<type> 可以取：

- **b** Boxcar: All weights are equal
- **c** Cosine Arch: Weights follow a cosine arch curve
- **g** Gaussian: Weights are given by the Gaussian function
- **f** Custom: Instead of *width* give name of a one-column file with your own weight coefficients.

对于非卷积滤波器，<type> 可以取：

- **m** Median: Returns median value
- **p** Maximum likelihood probability (a mode estimator): Return modal value. If more than one mode is found we return their average value. Append - or + to the filter width if you rather want to return the smallest or largest of the modal values.
- **l** Lower: Return the minimum of all values.

- L Lower: Return minimum of all positive values only.
- u Upper: Return maximum of all values.
- U Upper: Return maximum or all negative values only.

大写的 B|C|G|M|P|F will use robust filter versions, i.e., replace outliers (2.5 L1 scale off median) with median during filtering.

In the case of L|U it is possible that no data passes the initial sign test; in that case the filter will return 0.0.

-I<ignore\_val> 若输出数据等于 <ignore\_val> , 则设置其值为 NaN

-L<lack\_width> 检查数据间断。若输入数据存在超过 <lack\_width> 的间断, 则该数据点不输出值。

-N<t\_col> 指定哪一列数据包含自变量 (即时间)。默认值为 0, 即第一列。

-Q<q\_factor> 检查输出值的质量因子。

<q\_factor> 的取值为 0 到 1, 若某点的卷积的平均权重小于 <q\_factor> , 则该点的输出会被压制。

-S<symmetry\_factor> 检查数据关于时间窗中心的对称性。

<symmetry\_factor> 的取值范围为 0 到 1。若  $(\text{abs}(n\_left - n\_right)) / (n\_left + n\_right) > \text{factor}$ , 则该点不输出值。

-T<t\_min>/<t\_max>/<t\_inc>[+] 见官方文档

## 示例

```
gmt filter1d cruise.gmtd -T0/1.0e5/2000 -FG10000 -N3 -V > filtered_cruise.gmtd
```

```
gmt filter1d v3312.dt -FM50 -T0/100000/25 -L10 -S0.3 > v3312_filt.dt
```

---

## fitcircle

---

官方文档 [fitcircle](#)

**简介** 拟合球面上数据点的平均位置及圆弧

该命令或从输入数据的前两列读取经纬度数据，并计算所有坐标点的平均位置以及可以拟合这些坐标点的大圆路径的 pole。在计算时有两种计算方法，分别用 `-L1` 和 `-L2` 表示。

## 选项

`<table>` 输入数据

`-L<norm>`

- `-L1` 解法 1，详见官方文档
- `-L2` 解法 2，详见官方文档
- `-L` 或 `-L3` 同时输出解法 1 和解法 2 的结果

`-Ff|m|n|s|c` 控制输出格式。

正常情况下，该命令会将计算结果以较复杂的形式输出。使用 `-F` 选项，则只返回简单的坐标。`-F` 后可以加上其他修饰符以指定要返回的坐标：

- `f` Flat Earth mean location
- `m` mean location
- `n` north pole of great circle
- `s` south pole of great circle
- `c` pole of small circle and its colatitude, which requires `-S`

`-S[<lat>]` 拟合小圆弧而不是大圆弧，见官方文档

## 示例

如下命令，用两种计算方法拟合了数据的大圆弧路径和小圆弧路径，并借助`project`生成路径坐标。

```
#!/bin/bash
# test/fitcircle/circles.sh
#
# Tests gmt fitcircle on spherical data points

ps=fitcircle_ex1.ps

# Fit great circle to red points and small circle to green points
# Plot the best-fitting great and small circles as well as the
# location of the mean locations and pole locations for both
# the -L1 and -L2 options (i.e., -L3).

gmt fitcircle gcircle.txt -L3 > g.txt
gmt fitcircle scircle.txt -L3 -S > s.txt
gpole1=`grep "L1 N Hemisphere" g.txt | gawk '{printf "%s/%s\n", $1, $2}'`
gpole2=`grep "L2 N Hemisphere" g.txt | gawk '{printf "%s/%s\n", $1, $2}'`
spole1=`grep "L1 Small Circle Pole" s.txt | gawk '{printf "%s/%s\n", $1, $2}'`
spole2=`grep "L2 Small Circle Pole" s.txt | gawk '{printf "%s/%s\n", $1, $2}'`
slat1=`grep "L1 Small Circle" s.txt | gawk '{print 90-$NF}'`
slat2=`grep "L2 Small Circle" s.txt | gawk '{print 90-$NF}'`
gmt psxy -Rg -JG-30/40/7i -P -Bg -K gcircle.txt -Sc0.04i -Gred -Xc -Yc > $ps
gmt psxy -R -J -O -K scircle.txt -Sc0.04i -Ggreen >> $ps
gmt project -G1 -T$gpole1 -L-180/180 | gmt psxy -R -J -O -K -W3p >> $ps
gmt project -G1 -T$gpole2 -L-180/180 | gmt psxy -R -J -O -K -W1p,- >> $ps
gmt project -G1/$slat1 -T$spole1 -L-180/180 | gmt psxy -R -J -O -K -W3p >> $ps
gmt project -G1/$slat2 -T$spole2 -L-180/180 | gmt psxy -R -J -O -K -W1p,- >> $ps
grep "Great Circle Pole" g.txt | gmt psxy -R -J -O -K -Sa0.2i -Gred -W0.25p >> $ps
grep "Small Circle Pole" s.txt | gmt psxy -R -J -O -K -Sa0.2i -Ggreen -W0.25p >> $ps
grep "L1 Average" g.txt | gmt psxy -R -J -O -K -Sa0.2i -Gyellow -W0.25p >> $ps
grep "L2 Average" g.txt | gmt psxy -R -J -O -K -Sa0.2i -Gyellow -W0.25p >> $ps
grep "L1 Average" s.txt | gmt psxy -R -J -O -K -Sa0.2i -Gyellow -W0.25p >> $ps
grep "L2 Average" s.txt | gmt psxy -R -J -O -K -Sa0.2i -Gyellow -W0.25p >> $ps
gmt psxy -R -J -O -T >> $ps
rm g.txt s.txt gmt.*
```

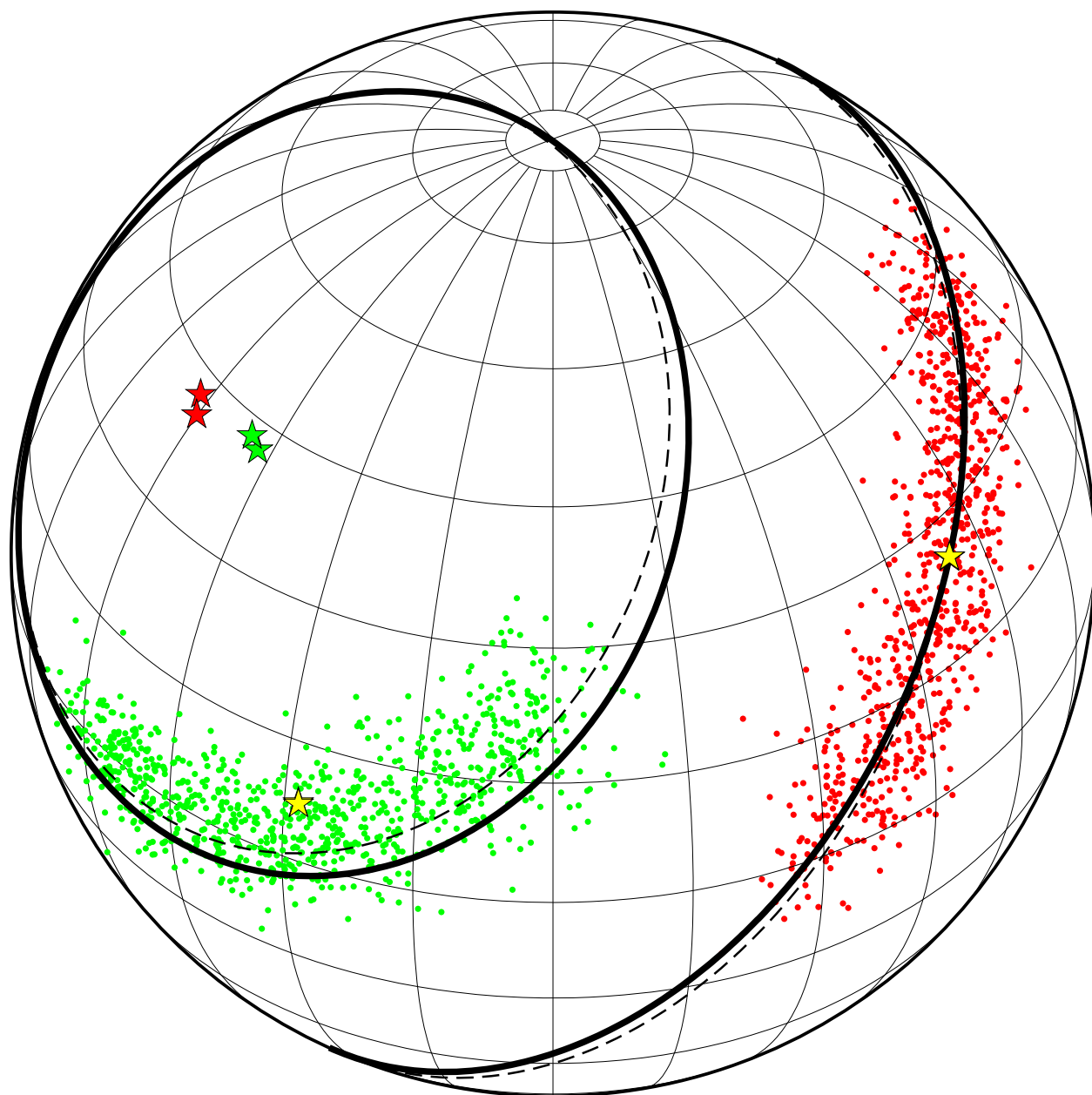


图 7.1: fitcircle 示例

---

## gmt

---

### 官方文档 [gmt](#)

**说明** GMT 的主程序，也是 GMT5 中唯一的一个二进制文件，所有的模块都要通过该命令调用。

GMT 的所有功能通过如下语法调用：

```
gmt <module> <module-options>
```

其中，<module> 是 GMT 模块名，<module-options> 是模块所指定的选项。

除此之外，还有一些其他选项可以使用：

- `gmt --help` : 列出 GMT 的模块列表及其描述
- `gmt --version` : 显示 GMT 版本
- `gmt --show-bindir` : 显示 GMT 的 bin 目录
- `gmt --show-datadir` : 显示 GMT 的数据目录，默认为空
- `gmt --show-sharedir` : 显示 GMT 的 share 目录
- `gmt --show-plugindir` : 显示 GMT 插件目录
- `gmt --show-modules` : 列出 GMT 的所有模块名
- `gmt --show-cores` : 显示当前计算机可以使用的核数
- `gmt <module> =` : 检测模块 <module> 是否存在，若存在则返回 0，否则返回非零值

---

**gmt2kml**

---

官方文档 [gmt2kml](#)



---

## gmt5syntax

---

官方文档 [gmt5syntax](#)

**简介** 将旧的 GMT 脚本修改成新的 `gmt <module>` 语法。

`gmt5syntax` 是一个 Perl 脚本，作用是在命令前加上 `gmt` 使得旧版本的语法变成新版本语法。该脚本位于 `${GMTHOME}/share/tools` 目录。

## 语法

```
gmt5syntax old_script > new_script
```

---

## gmtconnect

---

### 官方文档 [gmtconnect](#)

**简介** 将端点接近的线段连接起来

该命令会读入一个或多个多段数据，并检查所有线段的两个端点。若某对端点完全相同或者二者的距离在可容忍的范围内，则将两段线段连接成一条线段。该进程会一直重复，直到剩余的端点均不在可容忍的范围内。最终连接得到的线段会被写到终端或特定的输出文件中。

若不清楚该如何确定可容忍范围值，可以使用 `-L` 选项，获得所有端点之间的距离的列表，通过分析该列表来得到合适的值。

## 选项

`-C[<closed>]` 将所有闭合多边形写到文件 `<closed>`（默认文件名为 `gmtconnect_closed.txt`）中并将其他数据段写到标准输出。

使用该选项不会对线段做连接。

`-D[<template>]` 对于多段数据文件，将每段数据分别输出到不同的数据文件中。

`<template>` 是文件名的模板，该模板中必须包含一个整型参数的格式，比如 `%d` 或 `%08d`，也可以在整型参数格式前加上字符参数格式 `%c`，实际输出时会被 `C` 或 `O` 代替，分别表示 `closed` 和 `open`。

默认的模板为 `gmtconnect_segment_%d.txt`。

`-L[<linkfile>]` 将连接信息写到指定的文件中，默认文件名为 `gmtconnect_link.txt`。

对于每段数据而言，会写入原始的数据段 ID；对于线段的起始点和终点而言，会报告离得最近的线段的 ID，以及两个线段端点之间的距离。

`-Q[<template>]` 见官方文档

`-T[<cutoff>[<unit>]][/<nn_dist>]` 若两条线段的端点的距离小于 `<cutoff>` 则连接这两条线段。

`<nn_dist>` 见官方文档。

## 示例

```
gmt connect segment_*.txt -Tf0.1 > new_segments.txt
```

```
gmt connect my_lines.txt -T150e -DMap_segment_%04d.dat
```

---

**gmtconvert**

---

官方文档 [gmtconvert](#)

---

# gmtdefaults

---

官方文档 [gmtdefaults](#)

简介 列出所有 GMT 参数的当前值

## 选项

-D 打印系统默认参数

1. -D : 列出 `${GMT}/share/conf/gmt.conf` 中的参数值
2. -Du : 列出 `${GMT}/share/conf/gmt_US.conf` 中的参数值
3. -Ds : 列出 `${GMT}/share/conf/gmt_SI.conf` 中的参数值

## BUGS

- -D 选项无效 (v5.1.2)

---

**gmtflexure**

---

官方文档 [gmtflexure](#)

---

## gmtget

---

官方文档 [gmtget](#)

简介 列出单个或多个 GMT 参数的当前值

## 选项

<PARAMETER> GMT 的参数名，见 [configurations](#)。

gmtset 后直接跟一个或多个参数名：

```
$ gmt gmtget PS_MEDIA
a4

$ gmt get MAP_GRID_CROSS_SIZE_PRIMARY MAP_GRID_CROSS_SIZE_SECONDARY
24p,Helvetica,black 16p,Helvetica,black fancy
```

-G<defaultsfile> 读取指定的配置文件

该命令寻找配置文件 `gmt.conf` 时的搜索优先顺序为：

```
./gmt.conf > ~/.gmt/gmt.conf > ~/gmt.conf > ${GMTHOME}/share/conf/gmt.conf
```

-L 输出时一行只输出一个返回值。

一次指定多个参数名时，默认会将所有返回值输出在一行，各值之间以空格分隔。该选项会一行只输出一个返回值。

```
$ gmt get FONT_TITLE FONT_LABEL MAP_FRAME_TYPE -L
24p,Helvetica,black
16p,Helvetica,black
fancy
```

---

**gmtgravmag3d**

---

官方文档 [gmtgravmag3d](#)



---

## gmtinfo

---

官方文档 [gmtinfo](#)

简介 从表数据中提取极值信息

### 最小示例

提取文件中各列的极值:

```
$ gmt info input.dat
input.dat: N = 6      <1/2>   <1/3>   <0/3.2>
```

从输出中可以看到，输入文件中有 6 行 3 列，三列数据的范围分别是 1 到 2、1 到 3 和 0 到 3.2。

### 选项

**-Aa|f|s** 输入数据为多段数据或多文件时的处理方式:

1. **-Aa** : 报告所有文件的总最值
2. **-Af** : 分别报告各个文件自己的最值
3. **-As** : 分别报告各段数据自己的最值

**-C** 输出时，每列的最大最小值分别占一列

在输出每一列的最大最小值时，默认会用 `<min/max>` 的格式输出，不易于脚本处理。该选项会将每列的最大最小值分别输出为一列:

```
$ gmt info input.dat -C
1  2  1  3  0  3.2
```

该选项经常与 **-i** 及 **-o** 选项一起使用，以提取部分列的最值信息。

-D 对 -I 选项的结果做整体偏移，使得区域范围的中心与原数据的中心对齐。

比如数据的真实范围是 0/3，使用 -I2 选项会得到 0/4，在此基础上使用 -D 选项，则变成 -0.5/3.5。

-EL|l|H|h<col> 返回第 <col> 列最值所在的记录行。

1. l|h 表示返回最小、最大值所在的记录行
2. L|H 表示返回绝对值最小、最大值所在的记录行

若有多行均满足要求，则只返回第一行。若 <col> 未指定，则默认是最后一列。

-I 调整最大/小值使得其是给定的增量的最近倍数，返回的格式为 -Rw/e/s/n

```
$ gmt info input.dat -I2/2
-R0/2/0/4
```

如果设置 <dx> 的值为 -，则表示以 -Rw/e/s/n 的格式输出真实的 -R 范围：

```
$ gmt info input.dat -I-
-R1/2/1/3
```

当 -I 选项和 -C 选项一起使用时，输出不再是 -Rw/e/s/n 格式：

```
$ gmt info input.dat -C -I2
0  2  0  4  0  3.2
```

上面的例子中，-I 选项后只接了一个增量，此时第二列数据也会使用同样的增量（出于向后兼容性考虑）。若只想要为第一列指定增量，第二列不指定，可以使用 -Ip 选项：

```
$ gmt info input.dat -C -Ip2
0  2  1  3  0  3.2
```

对于某些算法，比如 FFT 或者 surface 命令，-R 选项的不同取值会影响到运算的速度，-If 会输出优化后的 -R 以得到最快 FFT 结果，-Is 会输出优化后的 -R 以得到最快的 surface 结果。

-S 见官方文档。

-T<dz>[/<col>] 以 -Tzmin/zmax/dz 的形式报告第 <col> 列的最值，默认报告第 1 列。该选项必须 -I 选项连用。

## gmtlogo

官方文档 [gmtlogo](#)

**说明** 在图上绘制 GMT 的图形 logo

该模块将 GMT 的图形 logo 绘制在图上。GMT 的图形 logo 默认宽 2 英寸，高 1 英寸。

## 选项

**-D**[g|j|J|n|x]<refpoint>+w<width>[+j<justify>][+o<dx>[/<dy>]] 设置 logo 的位置。

1. [g|j|J|n|x]<refpoint> 设置底图上的参考点，[embellishments](#) 一节
2. +j<justify> 设置 logo 上的锚点，见 [embellishments](#) 一节
3. +o<dx>[/<dy>] 设置参考点的额外偏移量，见 [embellishments](#) 一节
4. +w<width> 设置 logo 的宽度

**-F** 设置 logo 的背景属性，见 [embellishments](#) 一节

## 示例

单独绘制 GMT logo:

```
gmt logo -P -Dx0/0+w2i > logo.ps
```

将 GMT logo 作为图片的一个图层:

```
gmt logo -O -K -R -J -DjTR+o0.1i/0.1i+w3i >> bigmap.ps
```

---

**gmtmath**

---

官方文档 [gmtmath](#)

---

**gmtregress**

---

官方文档 [gmtregress](#)

---

## gmtselect

---

### 官方文档 [gmtselect](#)

#### 简介 根据多个准则筛选数据

该命令会从输入文件中读取前两列作为经度和纬度，并判断经纬度是否符合一定的空间准则，以筛选出符合条件的记录。输入文件中仅前两列会被使用。

六个空间准则包括：

1. 在矩形区域内（`-R` 和 `-J`）
2. 与 `ptfile` 中的每个点的距离在 `dist` 之内
3. 与 `linefile` 中的每条线的距离在 `dist` 之内
4. 在 `polygonfile` 中的多边形内
5. 在某个地理区域内（需要海岸线数据）
6. `z` 值在某个范围内

## 六个空间准则

**准则 1** 筛选出在某个矩形区域内（通过 `-R` 和 `-J` 选项控制）的点：

```
gmt select points.xy -R0/5/0/5
```

**准则 2** 与 `<ptfile>` 中的每个点的距离在 `<dist>` 内，由 `-C` 选项控制，其语法为：

```
-C<dist>[<unit>]/<ptfile>
```

该准则会筛选出与文件 `<ptfile>` 中的每个点的距离在 `<dist>` 之内的记录。若 `dist` 等于 0，则 `<ptfile>` 中的第三列是每个数据点各自的影响半径，即筛选出不在任何一个数据点的影响半径内的点。

默认情况下 `<dist>` 是笛卡尔坐标系下的距离，单位为用户单位。若指定 `-fg` 选项，则表明 `<dist>` 为球面距离。若使用了 `-R` 和 `-J`，则 `<dist>` 表示投影后的纸面距离。

**准则 3** 与 `linefile` 中的每条线的距离在 `dist` 之内，由 `-L` 选项控制，其语法为：

```
-L[p]<dist>[<unit>]/<linefile>
```

`<linefile>` 中包含了一系列线段，该准则会筛选出与这些线段的距离不超过 `<dist>` 的记录。

若 `<dist>` 等于零，则可以在 `<ptfile>` 中每段数据的段头记录中使用 `-D<dist>` 参数，为每个线段分别指定距离值。

默认情况下 `<dist>` 是笛卡尔坐标系下的距离，单位为用户单位。若指定 `-fg` 选项，则表明 `<dist>` 为球面距离。若使用了 `-R` 和 `-J`，则 `<dist>` 表示投影后的纸面距离。

使用 `-Lp` 则会将数据点垂直投影到线段上，只有投影的位置在线段的端点内的记录才会被保留，即只有线段左右一定距离内的点才会被保留，线段两端点一定距离内的点不会被保留。

**准则 4** 筛选出在某个多边形内的点，由 `-F` 选项控制，其语法为：

```
-F<polygonfile>
```

`<polygonfile>` 中可以包含一个或多个多边形，该选项筛选出所有在多边形内的记录。

**准则 5** 根据地理特征信息筛选数据，由 `-N` 和 `-D` 选项控制。`-N` 选项的语法为：

```
-N<wet>/<dry>
```

或：

```
-N<ocean>/<land>/<lake>/<island>/<pond>
```

其中每一项都可以取为 `slk`，分别表示 `skip` 或 `keep`，即跳过或保留该地理特征内的记录。默认值为 `-Ns/k`，即保留所有陆地上的记录。

**准则 6** 筛选 `Z` 值在某个范围内的点，由 `-Z` 选项控制。其语法为：

```
-Z<min>[/<max>][+c<col>]
```

判断记录的 `Z` 值是否在 `<min>` 到 `<max>` 之间。可以使用 `+c<col>` 指定记录中的某一列作为 `Z` 值，默认以第三列（`col=2`）作为 `Z` 值。若省略了 `<max>`，则判断 `Z` 值是否等于 `<min>`。若不限制范围的上限或下限，可以使用 `-` 代替。

若想要对多列做类似的测试，可以重复使用 `-Z` 选项，每次指定不同的列号。

## 选项

**-A** 不考虑面积过小的湖泊的边界，或不考虑某个级别的湖泊边界。见 [pscoast](#) 中 `-A` 选项的介绍。

- D<resolution>[+]** 选择海岸线数据的精度，仅与 **-N** 选项一起使用有效。见 [pscoast](#) 中 **-D** 选项的介绍。
- E[fn]** 在判断点是否在一个多边形内时，默认会将恰好在多边形上的点也认为是在多边形内，使用该选项会将多边形上的点认为是在多边形外部。**f** 和 **n** 分别对应于 **-F** 和 **-N** 选项。
- I[cflrsz]** 对六个准则取反，即筛选出不符合准则的记录。**c**、**f**、**l**、**r**、**s**、**z** 分别对应于 **-C**、**-F**、**-L**、**-R**、**-S** 和 **-Z**

## 示例

筛选与 **pts.d** 中所有点的距离在 300 km 以内，与 **lines.d** 中线段的距离在 100 km 以外的点：

```
gmt select lonlatfile -fg -C300k/pts.d -L100/lines.d -Il > subset
```

此处需要使用 **-fg** 以告知程序正在处理地理数据。

筛选某个区域内所有不在陆地上的点：

```
gmt select data.d -R120/121/22/24 -Dh -Nk/s > subset
```

**stations.d** 中的点投影后与 **origin.d** 的距离在 5 cm 之内的点：

```
gmt select stations.d -C5/origin.d -R20/50/-10/20 -JM20c \  
--PROJ_LENGTH_UNIT=cm > subset2
```



---

## gmtset

---

### 官方文档 [gmtset](#)

**简介** 修改单个或多个 GMT 参数的值

该命令会修改当前目录下 GMT 配置文件 `gmt.conf` 中指定参数的值。若配置文件不存在，则新建一个。由于该目录会修改当前目录下的配置文件 `gmt.conf`，因而该命令修改的参数将对接下来的命令一直有效，直到 GMT 参数再次被修改或覆盖。

若想要参数修改仅对某个命令有效，应该在该命令上使用 `--<PARAMETER>=<value>` 语法。

`<PARAMETER>=<value>` 要修改的 GMT 参数名 `<PARAMETER>` 以及想要设置的值 `<value>`

参数名和值必须成对存在，二者可以用等号 `=` 连接，也可以省略等号。

修改主标注字体 Helvetica，字号为 12p，设置网格交叉线的尺寸为 0.2 厘米：

```
gmt gmtset FONT_ANNOT_PRIMARY 12p,Helvetica MAP_GRID_CROSS_SIZE_PRIMARY 0.2c
```

也可以用等号将参数名和参数值连接起来：

```
gmt gmtset FONT_ANNOT_PRIMARY=12p,Helvetica MAP_GRID_CROSS_SIZE_PRIMARY=0.2c
```

**-C** 将当前目录下的 GMT4 配置文件 `.gmtdefaults4` 转换为 GMT5 所使用的 `gmt.conf` 文件，并保留 GMT4 配置文件。

**-D[s|u]** 默认在当前目录下的配置文件的基础上修改参数值，该选项使得在系统配置文件的基础上修改参数值。

1. **-D** : 使用 GMT 编译过程中指定的默认参数文件，即 `${GMT}/share/conf/gmt.conf`
2. **-Du** : 使用 US 单位制下的默认参数文件，即 `${GMT}/share/conf/gmt_US.conf`
3. **-Ds** : 使用 SI 单位制下的默认参数文件，即 `${GMT}/share/conf/gmt_SI.conf`

使用 **-D** 选项而不指定参数名则会在当前目录下生成一个系统配置文件：

```
gmt set -D
```

**-G<defaultsfile>** 指定要读取并修改的配置文件，<defaultsfile> 为配置文件的文件名

**-[BJRXxYycp]<value>** 修改 GMT 命令历史文件 `gmt.history` 中的值。该选项用于在不执行其他绘图命令的前提下强行修改命令历史，但用处不大。

例如：

```
$ gmt psbasemap -JX10c/5c -B0/10/0/10 -B1 > test.ps
$ gmt gmtset -JX5c/5c
```

该命令会将 GMT 命令历史文件 `gmt.history` 中 `-J` 选项的值从 `X10c/5c` 改成 `X5c/5c` 。

## FAQ

### 1. 错误消息：

```
gmtset: Warning: parameter xxxx is deprecated. Use xxx instead.
```

出现该错误是因为 GMT5 重命名了几乎所有参数，而当前命令使用的是 GMT4 的老参数名。解决办法在警告信息中已经说了，使用 GMT5 的新参数名即可。

---

## gmtsimplify

---

官方文档 [gmtsimplify](#)

**简介** 使用 Douglas-Peucker 算法对线段做简化

该命令可以将一个复杂多边形进行简化，用一条直线代替一系列点，并保证每个点与直线的偏离都在可容忍的范围内。

### 选项

**-T<tolerance>[<unit>]** 指定最大所能容忍的误差。默认单位为用户单位，也可以指定其他距离单位。

### 示例

将线段简化，可容忍误差为 2 千米：

```
gmt simplify segment.d -T2k > new_segment.d
```

---

**gmtspatial**

---

官方文档 [gmtspatial](#)

---

**gmtswitch**

---

官方文档 [gmtswitch](#)

---

**gmtvector**

---

官方文档 [gmtvector](#)

---

## gmtwhich

---

官方文档 [gmtwhich](#)

**简介** 返回指定文件的完整路径

GMT 会依次在三个目录中去寻找命令行中指定的文件:

当前目录 > `$GMT_USERDIR` > `$GMT_DATADIR`

该命令会报告文件的完整路径，使得用户可以确认自己在使用的究竟是哪个数据文件。

## 选项

**<files>** 任意数据文件名

- A 仅考虑用户有读权限的文件
- C 不报告完整路径，只打印 Y 或 N 以表示是否找到文件
- D 不报告完整路径，仅打印包含该文件的目录名

---

**gravfft**

---

官方文档 [gravfft](#)



---

**grd2cpt**

---

官方文档 [grd2cpt](#)

---

## grd2rgb

---

### 官方文档 [grd2rgb](#)

**简介** 将网格文件、原始 RGB 文件、SUN 光栅文件转换成 R/G/B 三个网格文件

该命令的作用是：

1. 读取 Sun 光栅文件（8、24 或 32 位），并将其中的 red、green、blue 分量分别写到单独的网格文件中
2. 读取 2D 网格文件，并根据指定的 CPT 文件，将 Z 值转换为对应的 RGB 值，然后将 red、green、blue 分量分别写到单独的网格文件中
3. 读取 RGB 或 RGBA 原始光栅文件，并将 red、green、blue 分量分别写到单独的网格文件中（需要使用 -W 选项）

## 选项

**<infile>** 输入文件，可以是 Sun 光栅文件、2D 网格文件或原始光栅文件

**-G<template>** 要生成的三个网格文件的文件名模板，模板中必须包含 %c

例如，-Gout\_%c.nc 会生成三个网格文件 out\_r.nc、out\_g.nc 和 out\_b.nc

**-C<cpt>** 若输入文件是 2D 网格文件，则需要指定 CPT 文件，以将 Z 值转换为 RGB 颜色

**-I<xinc>[<unit>][=|+]/<yinc>[<unit>][=|+]** 指定 X 和 Y 方向的网格间隔，见 [xyz2grd](#) 中的介绍

**-L<layer>** 只输出某一分量，<layer> 可以取 r|g|b 。默认会输出三个分量。

**-W<width>/<height>[/<n\_bytes>]** 设置原始光栅文件中的尺寸信息。

原始光栅文件中不包含图片的尺寸信息，因而需要单独的选项来设置。默认情况下，假定光栅文件是 RGB 原始文件，即每个像素 3 个字节；对于 RGBA 文件，每个像素 4 个字节，此时 <n\_bytes>=4 。

使用 -W=/=/4 可以自动猜测 RGBA 光栅文件的尺寸信息。

## 示例

```
gmt grd2rgb hawaii_grv.nc -Ctopo.cpt -Ghawaii_grv_%c.nc
```

```
gmt grd2rgb radiation.ras -Lr -Gcomp_%c.nc
```

---

## grd2xyz

---

官方文档 [grd2xyz](#)

**简介** 将网格文件转换成表数据

该命令或读取一个或多个 2D 网格文件，并将 XYZ 数据以 ASCII 或二进制格式写到标准输出中。ASCII 输出的格式由参数 `FORMAT_FLOAT_OUT` 控制。

## 选项

**<grid>** 要转换的 2D 网格文件

**-C[f|i]** 输出的 X、Y 坐标用对应的列、行号替代

默认输出的三列数据是：X 坐标、Y 坐标和 Z 值。使用该选项，则输出的三列数据为：列号、行号和 Z 值。其中，行号和列号从 0 开始算起。使用 **-Cf** 则行号和列号从 1 开始算起。**-Ci** 会输出两列数据：索引和 Z 值。索引相当于是将二维数组用一维数组表示。

**-R** 指定要转换的网格文件的数据范围

**-W[<weight>]** 默认输出 XYZ 三列数据，使用该选项则输出四列 XYZW，其中 W 即 `<weight>`

**-Z[flags]** 输出一列 ASCII 表或二进制表。

使用该选项，则输出时只有 Z 值，没有 XY 信息。输出 Z 值的顺序由 `<flags>` 决定。若是行优先，`<flags>` 的第一个字符可以取：

- T 表示第一行是 `y=ymax`
- B 表示第一行是 `y=ymin`

`<flags>` 的第二个字符可以取：

- L 表示每一行的第一个元素是 `x=xmin`
- R 表示每一行的第一个元素是 `x=xmax`

若是列优先，则 **L|R** 为第一个字符，**B|T** 为第二个字符。

对于网格线配准的网格文件而言：

- 若网格在 X 方向是周期的，输出数据时不需要包含  $x=x_{\max}$  所在的列，则加上 **x**
- 若网格在 Y 方向是周期的，输出数据时不需要包含  $y=y_{\max}$  所在的行，则加上 **y**

若数据需要做字节交换，则加上 **w** 。最后需要指定，数据以何种数据类型保存：

- **a** ASCII 表，每行输出一个 Z 值
- **c** int8\_t, signed 1-byte character
- **u** uint8\_t, unsigned 1-byte character
- **h** int16\_t, short 2-byte integer
- **H** uint16\_t, unsigned short 2-byte integer
- **i** int32\_t, 4-byte integer
- **I** uint32\_t, unsigned 4-byte integer
- **l** int64\_t, long (8-byte) integer
- **L** uint64\_t, unsigned long (8-byte) integer
- **f** 4-byte floating point single precision
- **d** 8-byte floating point double precision

默认值为 **-ZTLa** 。

## grdblend

官方文档 [grdblend](#)

简介 将多个部分重叠的网格文件合并成一个网格文件

## 选项

**<blendfile>** 文件中包含了要合并的网格文件的文件名列表，其格式为:

网格文件名（必须）    **-R** 选项（可选）    相对权重（可选）

- 在考虑权重的情况下，**-R** 范围外的部分会给零权重，范围内的部分则会使用指定的相对权重，在边界处会加上 2D 余弦 taper 权重。
- 若相对权重为负值，则整个反过来，即范围内的会给零权重，范围外的会给权重的绝对值。
- **-R** 选项可以取值 **-**，此时会直接使用网格文件的数据范围。
- 若未指定权重，则默认权重为 1
- 也可以不指定 **<blendfile>**，只是将所有网格文件名列在命令行中，此时会使用网格数据的真实范围，且所有网格文件的权重都是 1.

**-C[f|l|o|u]** 不考虑权重

- **-Cf** 将第一个访问该节点的网格数据作为该节点的值
- **-Co** 将最后一个访问该节点的网格数据作为该节点的值
- **-Cl** 将所有网格文件在该节点的值的的最小值作为该节点的值
- **-Cu** 将所有网格文件在该节点的值的最大值作为该节点的值

**-G<outgrid>** 输出的网格文件名

**-I<xinc>[<unit>][=|+]/<yinc>[<unit>][=|+]** 指定 X 和 Y 方向的网格间隔

- **<xinc>** X 方向的网格间隔

- `<yinc>` Y 方向的网格间隔
- `<unit>` 网格间隔的单位
- `=` 微调 X 和 Y 方向范围的最大值，使得其是网格间隔的整数倍（默认会微调网格间隔以适应给定的数据范围）
- `+` 表明 `<xinc>` 和 `<yinc>` 不是网格间隔，而是 X 和 Y 方向的节点数

`-N<nodata>` 将无数据的节点的值设置为 `<nodata>`，默认值为 NaN

`-Q` 创建一个无头段的二进制网格文件，使得其可以被 *grdraster* 所用。

`-W[z]` 不合并数据，仅输出每个节点所使用的权重。加上 `z` 则输出权重乘以 Z 值的和。

`-Z<scale>` 在输出前先将数据乘以比例因子 `<scale>`

---

## grdclip

---

官方文档 [grdclip](#)

简介 对网格文件的 Z 值做裁剪

### 选项

<ingrid> 输入 2D 网格文件

-G<outgrid> 输出网格的文件名

-R 指定要截取的网格区域

-S 裁剪参数，该选项可以多次使用。

- -Sa<high>/<above> 将所有大于 <high> 的值设置为 <above>
- -Sb<low>/<below> 将所有小于 <low> 的值设置为 <below>
- -Si<low>/<high>/<between> 将所有在 <low> 和 <high> 范围内的值设置为 <between>，该选项可多次使用
- -Sr<old>/<value> 将所有等于 <old> 的值设置为 <new>，该选项可以多次使用

### 示例

将所有大于 70 的值设置为 NaN，并将小于 0 的值设置为 0:

```
gmt grdclip data.nc -Gnew_data.nc -Sa70/NaN -Sb0/0 -V
```

将所有 25 到 30 范围内的值设置为 99，35 到 39 范围内的值设置为 55，将 17 换成 11，将所有小于 10 的值设置为 0:



```
gmt grdclip classes.nc -Gnew_classes.nc -Si25/30/99 -Si35/39/55 -Sr17/11 -Sb10/0 -V
```

---

**grdcontour**

---

官方文档 [grdcontour](#)

## grdcut

官方文档 [grdcut](#)

**简介** 从一个网格文件中裁剪出一个子区域

可以通过指定 **-R** 直接指定自区域的范围，也可以通过检查网格节点的值或者离指定点的距离来限定子区域。

## 选项

**<ingrid>** 输入网格文件

**-G<outgrid>** 输出网格文件名

**-N[<nodata>]** 允许新网格的范围超过原网格范围。

一般情况下，若指定的区域范围大于输入网格的区域范围，超出的部分会被自动忽略，实际的输出网格的区域范围会自动适应输入网格的区域范围。使用 **-N** 选项，则超出的区域范围内，网格节点会被赋以指定的值，默认赋值为 NaN，也可以自己指定其值 **<nodata>**。

**-R** 指定要截取的网格区域：

```
gmt grdcut input.nc -Goutput.nc -R0/30/-30/30
```

**-S[n]<lon>/<lat>/<radius>[<unit>]** 指定一个圆形区域，程序会自动计算出一个包含了圆上及圆内所有节点的矩形区域。

1. **<lon>** **<lat>** 圆心位置
2. **<radius>** 圆半径
3. **<unit>** 半径单位
4. **-Sn** 设置矩形区域内位于圆外的节点的值为 NaN

下面的例子生成的网格文件包含了距离 (45,30) 500 km 以内的所有点：

```
gmt grdcut input.nc -Goutput.nc -S15/0/500k -V
```

**-Z[n|r]<min>/<max>** 定义一个 Z 值范围 [min,max]，由此确定一个新的子区域，保证在该子区域外的所有节点都位于给定的 Z 值范围外。注意：矩形区域内的网格点的值并不一定都在 Z 值范围之内。

**<min>** 和 **<max>** 默认值为正负无穷，实际使用时可以用 - 代替。

为了得到满足要求的矩形区域，代码中从四条边界开始向内收缩，当某条边界遇到某个网格点的值在给定的 Z 值范围内，则此边界停止收缩。

正常情况下，值为 NaN 的记录会被直接跳过：

- **-Zn** 认为 NaN 记录位于 Z 值范围之外，故而保证子区域内是没有 NaN 记录的
- **-Zr** 认为 NaN 记录位于 Z 值范围之内，因而当遇到 NaN 记录时，则停止收缩边界

```
gmt grdcut abc.nc -Gdef.nc -Z-/0 -V
```

## grdedit

官方文档 [grdraster](#)

简介 修改网格文件的头段或内容

## 语法

`<grid>` 要修改的网格文件

`-A` 如有必要，则对网格间隔做微调使得其与数据的范围相兼容。

`-D<xname>/<yname>/<zname>/<scale>/<offset>/<invalid>/<title>/<remark>` 修改网格文件的基本信息：

- `<xname>` X 变量名，格式为 `varname [unit]`，比如 `"distance [km]"`
- `<scale>` 读入网格数据后要乘以的因子
- `<offset>` 读入数据后并乘以因子后要加入的常数
- `<invalid>` 无有效值的节点处的替代值，默认为 NaN
- `<title>` 网格文件的标题
- `<remark>` 网格文件的注释信息

若想要某些值不变，可直接将该字段留空，但 `/` 不可省略。若 `/` 是某个值的一部分，则可以用其他符号作为分隔符，此时需要在开头和结尾多加一个该符号，比如 `-D:xname:yname:zname:scale:offset:invalid:title:remark:`。

假设数据的范围是 `300/310/10/30`，现修改数据的范围以及标题：

```
gmt grdedit data.nc -R-60/-50/10/30 -D////////"Gravity Anomalies"
```

`-E[a|h|l|r|t|v]` 对网格进行处理

- `-Ea` 将网格在 180 度附近旋转 (?)

- -Eh 水平旋转网格（从左到右）
- -El 逆时针将网格旋转 90 度
- -Er 顺时针将网格旋转 90 度
- -Et 对网格进行转置（想象成一个二维矩阵）
- -Ev 垂直旋转网格（从上到下）

-G<outgrid> 默认情况下，该命令会直接修改并覆盖原始网格文件，使用该选项则将修改后的网格写到新的文件中。

-N<table> 从文件 <table> 中读入数据，并用其替换网格中对应节点的值。

-R<w>/<e>/<s>/<n> 修改网格文件的范围。同时，网格间隔会做相应修改。

-S 仅用于全球地理网格数据。将网格沿着经度范围整体偏移，使得其满足 -R 定义的新的范围。

原数据范围是 0/360/-72/72，现将数据整体偏移 180 度使得数据范围是 -180/180/-72/72

```
gmt grdedit world.nc -R-180/180/-72/72 -S
```

-T 修改网格文件的头段，将一个网格线配置的文件变成像素配准的文件，或反之。

使用该选项后，网格线配准的数据的范围将在四个方向上扩大半个网格间隔，像素点配置的数据的范围将在四个方向上缩小半个网格间隔。

## 示例

假定数据范围是 300/310/10/30，下面的命令将范围修改为-60/50/10/30，并为其加上标题：

```
gmt grdedit data.nc -R-60/-50/10/30 -D////////"Gravity Anomalies"
```

网格范围为 0/360/-72/72，移动数据，使得其范围为-180/180/-72/72：

```
gmt grdedit world.nc -R-180/180/-72/72 -S
```

将网格数据逆时针旋转 90 度，并将旋转后的网格写到新网格文件中：

```
gmt grdedit oblique.nc -El -Goblique_rot.nc
```

---

**grdfft**

---

官方文档 [grdfft](#)

---

**grdfilter**

---

官方文档 [grdfilter](#)



---

**grdflexure**

---

官方文档 [grdflexure](#)

---

**grdgradient**

---

官方文档 [grdgradient](#)

---

**grdgravmag3d**

---

官方文档 [grdgravmag3d](#)

---

**grdhisteq**

---

官方文档 [grdhisteq](#)

---

## grdimage

---

官方文档 [grdimage](#)

简介 在地图上绘制网格数据

### 选项

**<grid>** 该命令的输入文件必须是 2D 网格文件，可以是一个只包含 Z 数据的网格文件，也可以是三个分别包含 R、G、B 值的网格文件。

**-A<out\_img>=<driver>** 将图片以光栅格式保存

默认图片会以 PostScript 代码的形式输出。若 GMT 支持 GDAL，则可以使用该选项，将图片以其他光栅格式保存。

1. **<out\_img>** 为要保存的文件名
2. **<driver>** 图片格式，见 GDAL 的文档

**-C<cpt>** 绘图时要使用的 CPT 文件名。

也可以直接使用 GMT 自带的 CPT 文件名，此时 GMT 会自动根据网格文件的 Z 值范围将自带的 CPT 采样成 16 级的连续 CPT 文件。也可以通过 **-C<color1>,<color2>[,<color3>,...]** 的语法构建一个线性连续 CPT 文件。

**-D** 表明输入的网格文件是需要通过 GDAL 读取的图片文件，见官方文档。

**-E[i|<dpi>]** 设置投影后网格的精度，默认值为 100。见官方文档

**-G[f|b]<color>** 见官方文档

**-I<intensfile>|<intensity>** 增加光照效果

可以给一个常数值，也可以给一个 Z 值范围在 -1 到 1 的网格文件，通常该网格文件用 **grdgradient** 生成。

**-M** 使用 YIQ 转换强制将其转换为灰度图。

- N 对于非矩形地图，在地图边界处不对图片做裁剪。
- Q 将值为 NaN 的节点处设置为透明色

---

## grdinfo

---

官方文档 [grdinfo](#)

简介 从 2D 网格文件中提取基本信息

能提取的信息包括：

- X、Y、Z 的最大和最小值
- 最大/最小 Z 值所在的位置
- X、Y 的网格间隔
- X 和 Y 方向节点数目
- 均值、标准差、中位数、L1 范数、值为 NaN 的节点数
- 网格配准方式

## 最小示例

`grdinfo` 后直接跟网格文件，会显示网格文件的信息：

```
$ gmt grdinfo test.nc
test1.nc: Title: ETOP05 global topography
test1.nc: Command: grdraster -R100/150/-30/30 1 -Gtest.nc
test1.nc: Remark: /opt/GMT-4.5.13/share/dbase/etopo5.i2
test1.nc: Gridline node registration used [Geographic grid]
test1.nc: Grid file format: nf = GMT netCDF format (32-bit float), COARDS, CF-1.5
test1.nc: x_min: 100 x_max: 150 x_inc: 0.0833333333333 name: longitude [degrees_east] nx: 601
test1.nc: y_min: -30 y_max: 30 y_inc: 0.0833333333333 name: latitude [degrees_north] ny: 721
test1.nc: z_min: -10376 z_max: 6096 name: m
test1.nc: scale_factor: 1 add_offset: 0
test1.nc: format: classic
```

从输出中可以看到很多信息：

- 网格文件中的标题信息；
- 生成该网格文件的命令；
- 网格文件的配准方式，此处为 Gridline 配准；
- 数据格式为 netCDF 32 位 float；
- 数据中 X 维度的最小值 x\_min、最大值 x\_max、网格间隔 x\_inc 以及数据点数 nx；
- 数据中 Y 维度的最小值 y\_min、最大值 y\_max、网格间隔 y\_inc 以及数据点数 ny；
- 数据中 Z 值的最小值 z\_min 和最大值 z\_max 以及其他信息；

## 选项

-C 将输出以 Tab 分隔显示在一行中。输出格式为：

```
name w e s n z0 z1 dx dy nx ny [x0 y0 x1 y1] [med scale] [mean std rms] [n_nan]
```

默认只输出前 11 列，其含义看名字就可以猜到了。若使用了 -M 、 -L1 、 -L2 、 -M 选项，则会输出对应的方括号内的项。

若使用了 -I 选项，则输出格式为：

```
NF w e s n z0 z1
```

其中 NF 是读入的网格文件数目。

```
$ gmt grdinfo test1.nc -C
test1.nc    100 150 -30 30  -10376  6096    0.0833333333333 0.0833333333333 601 721
```

-F 以另一种输出格式显示上例中的输出信息。

-I[<dx>[/<dy>]]r|b] 以 <dx> 和 <dy> 的整数间隔报告数据的范围

1. -I 后不接任何参数，会以 -I<xinc>/<yinc> 的格式输出网格间隔
2. -Ir 或 -I- 会以 -R<w>/<e>/<s>/<n> 的格式输出网格文件的真实范围
3. -Ib 会输出网格区域范围的四个顶点对应的坐标
4. -I<dx>/<dy> 会先获取网格的区域范围，并对该范围做微调使得其是 dx 和 dy 的整数倍，并以 -R<w>/<e>/<s>/<n> 的格式输出

```
$ gmt grdinfo test1.nc -I
-I0.0833333333333/0.0833333333333
$ gmt grdinfo test1.nc -Ir
-R100/150/-30/30
```



```
$ gmt grdinfo test1.nc -Ib
> Bounding box for test1.nc
100    -30
150    -30
150     30
100     30
$ gmt grdinfo test1.nc -I3/3
-R99/150/-30/30
```

当 `-I<dx>/<dy>` 与 `-C` 选项连用时:

```
$ gmt grdinfo test1.nc -I3/3 -C
1   99  150 -30 30 -10376 6096
```

**-L** 报告 Z 值的其他信息

- `-L0` : 扫描整个数据并报告 Z 值的范围, 而不仅仅是从网格的头段中读取
- `-L1` : 输出中位数以及 L1 scale( $L1\ scale = 1.4826 * Median\ Absolute\ Deviation$ )
- `-L2` : 输出均值、标准差以及均方根

**-M** 寻找并报告 Z 最小和最大值所对应的坐标, 以及值为 NaN 的网格点的数目

**-R** 从网格文件中取出一个子区域, 并报告该子区域的信息

**-T[s]<dz>** 以 `-Tzmin/zmax/dz` 的格式输出 Z 值范围

1. `-T<dz>` : 提取 Z 的最小最大值, 并做微调使得最值是 dz 的整数倍, 然后以 `-Tzmin/zmax/dz` 的格式输出
2. `-Ts<dz>` : 与上面类似, 唯一的区别在于会根据 Z 的绝对值最大值, 输出一个关于 0 对应的范围

```
$ gmt grdinfo test1.nc -T0.1
-T-10376/6096/0.1
$ gmt grdinfo test1.nc -Ts0.1
-T-10376/10376/0.1
```

## grdlandmask

官方文档 [grdlandmask](#)

**简介** 根据海岸线数据创建陆地-海洋的 mask 网格文件

该命令会根据海岸线数据，确定指定网格内的每个节点是在陆地上还是在水中，并给节点赋予不同的值。

### 选项

**-G<mask\_grd\_file>** 生成的 mask 网格文件的文件名

**-I<xinc>[<unit>][=|+]/<yinc>[<unit>][=|+]** 指定 X 和 Y 方向的网格间隔。见 [xyz2grd](#) 中的介绍。

**-A** 不考虑面积过小的湖泊的边界，或不考虑某个级别的湖泊边界。见 [pscoast](#) 中的介绍。

**-D<resolution>[+]** 选项海岸线数据的精度，见 [pscoast](#) 中的介绍。

**-N<maskvale>[o]** 设置网格节点的值，可以是数字，也可以是 NaN。o 表示将位于边界处的节点当做在边界的外部。该选项可以取两种格式：

1. -N<wet>/<dry>
2. -N<ocean>/<land>/<lake>/<island>/<pond>

默认值为 0/1/0/1/0 （即 0/1 ），即将水域内的网格设置为 0，将陆地内的网格设置为 1。

### 示例

将所有陆地上的节点设置为 NaN，水域上的节点设置为 1:

```
gmt grdlandmask -R-60/-40/-40/-30 -Dh -I5m -N1/NaN -Gland_mask.nc -V
```

生成全球 1x1 度的网格，并将不同性质的区域设置成不同的值:

```
gmt grdlandmask -R0/360/-90/90 -D1 -I1 -N0/1/2/3/4 -Glevels.nc -V
```

## grdmask

官方文档 [grdmask](#)

简介 根据多边形数据或点数据创建 mask 网格文件

## 选项

**<pathfiles>** 一个或多个 ASCII 数据文件，其中包含了多边形或数据点

**-G<mask\_grd\_file>** 生成的 mask 网格文件的文件名

**-I<xinc>[<unit>][=|+]/<yinc>[<unit>][=|+]** 指定 X 和 Y 方向的网格间隔

- **<xinc>** X 方向的网格间隔
- **<yinc>** Y 方向的网格间隔
- **<unit>** 网格间隔的单位
- **=** 微调 X 和 Y 方向范围的最大值，使得其是网格间隔的整数倍（默认会微调网格间隔以适应给定的数据范围）
- **+** 表明 **<xinc>** 和 **<yinc>** 不是网格间隔，而是 X 和 Y 方向的节点数

**-A[m|p|x|y]** 控制两点之间的连接方式，见 [psxy](#) 命令中对 **-A** 选项的介绍

**-N[z|Z|p|P]<values>** 设置位于多边形外部、边界和内部的节点值，默认值为 0/0/1，即多边形内部节点值为 1，其他节点值为 0。

**<values>** 的形式为 **<out>/<edge>/<in>**，可以是任意数值，也可以是 NaN。

- **-Nz** 将多边形内的节点设置为从多段数据的段头记录中获取的 Z 值，比如多边形段头记录中的 **-Z<zval>**、**-L<header>** 或 **-aZ=<name>**
- **-NZ** 与 **-Nz** 类似，只是其会将多边形的边界也当做是多边形的内部
- **-Np** 使用一个从 0 递增的数字作为多边形的 ID，也可以在其后加上一个数字以指定序列的起始值

- -NP 与 -Np 类似，只是其会将多边形的边界当做多边形的内部

需要注意，-Nz|Z|p|P 不能与 -S 连用。

-S<search\_radius>[<unit>] 对所有数据点设置一个搜索半径，设置圆内、圆边界、圆外部的节点值。

若 <search\_radius> 为 z ，则取输入数据的第三列作为半径。

若未使用 -S 选项，则认为输入数据是一个闭合的多边形。

## 示例

多边形内和边界上的节点值为 1，外部值为 0:

```
gmt grdmask coastline_*.xy -R-60/-40/-40/-30 -I5m -N1/0/0 -Gland_mask.nc=nb -V
```

数据点周围 50 千米范围的节点值为 1，其余为 0:

```
gmt grdmask data.xyz -R-60/-40/-40/-30 -I5m -NNaN/1/1 -S50k -Gdata_mask.nc=nb -V
```

将多边形的 ID 作为多边形内部节点的值:

```
gmt grdmask plates.gmt -R-40/40/-40/40 -I2m -Nz -Gplate_IDs.nc -aZ=POL_ID -V
```

将多边形的 ID 作为多边形内部节点的值，但多边形 ID 从 100 开始:

```
gmt grdmask plates.gmt -R-40/40/-40/40 -I2m -Np100 -Gplate_IDs.nc -V
```

---

**grdmath**

---

官方文档 [grdmath](#)

---

## grdpaste

---

官方文档 [grdpaste](#)

**简介** 将两个网格沿着其共同边界拼接成一个文件

要合并的两个网格文件必须拥有相同的网格间隔以及一条共同的边。若不满足，则需要使用[grdcut](#)、[grdsample](#) 命令对网格数据做处理再拼接。

## 示例

一个简单的示例:

```
gmt grdpaste file_a.nc file_b.nc -Goutfile.nc
```

对于地理网格而言，可能需要使用 `-f` 选项处理网格的经度周期性问题:

```
gmt grdpaste file_a.nc file_b.nc -Goutfile.nc -V -fg
```

---

**grdpmodeler**

---

官方文档 [grdpmodeler](#)



---

**grdproject**


---

官方文档 [grdproject](#)

**简介** 对网格数据做地图变换和逆变换

该命令可以将地理坐标下的网格数据投影到一个矩形网格中，也可以将一个矩形坐标系下的网格数据反投影到地理坐标下。

## 选项

**<in\_grdfile>** 要进行变换的 2D 网格数据

**-G<out\_grdfile>** 输出的网格文件名

**-C<dx>/<dy>** 默认投影后的坐标是相对于区域的左下角，该选项使得投影后的坐标相对于投影的中心。  
**<dx>/<dy>** 是要加到投影后坐标的偏移量。

**-D<xinc>[<unit>] [=|+] [/<yinc>[<unit>]] [=|+]** 指定新网格的网格间隔。

**-E<dpi>** 设置新网格的分辨率，即每英寸的点数。

**-F[c|i|p|e|f|k|M|n|u]** 见官方文档

**-I** 逆变换，将矩形区域变换成地理区域。

**-Mc|i|p** 见官方文档

## 选项

将地理网格数据变换成 Mercator 网格，分辨率为 300dpi:

```
gmt grdproject dbdb5.nc -R20/50/12/25 -Jm0.25i -E300 -r -Gdbdb5_merc.nc
```

将网格数据逆变换为地理网格:

```
gmt grdproject topo_tm.nc -R-80/-70/20/40 -Jt-75/1:500000 -I -D5m -V -Gtopo.nc
```

将 UTM（以米为单位）下的网格数据逆变换为地理网格:

```
gmt grdproject topo_utm.nc -R203/205/60/65 -Ju5/1:1 -I -Mm -Gtopo.nc -V
```

---

## grdraster

---

### 官方文档 [grdraster](#)

**简介** 从二进制数据中提取子区域并保存为 GMT 网格文件

该命令会读取根据 `grdraster.info` 的内容，读取指定的二进制数据，从中提取出一个区域里的数据，并保存为网格文件。

`grdraster.info` 位于 `${GMTHOME}/share/dbase` 目录下，文件内容的格式为：

文件号 文件标题 Z 值单位 -R 范围 -I 间隔 配准方式 数据格式 scale offset NaN 文件名

具体的细节在文件的注释中都有介绍。

该文件的作用在于，指定了本地二进制文件的基本信息，使得 GMT 可以精确地知道二进制数据的格式，进而可以准确读取二进制文件的内容。

## 选项

`<filename>` `grdraster.info` 中某文件所对应的文件号。或者也可以使用 `text pattern` 找到 `grdraster.info` 中数据描述中匹配的行。

`-G<grdfile>` 默认会直接将数据以 ASCII 表的形式输出到标准输出流。使用该选项，则会将数据写到网格文件中

`-I<xinc>[<unit>][=|+]/<yinc>[<unit>][=|+]` 指定 X 和 Y 方向的网格间隔

- `<xinc>` X 方向的网格间隔
- `<yinc>` Y 方向的网格间隔
- `<unit>` 网格间隔的单位
- `=` 微调 X 和 Y 方向范围的最大值，使得其是网格间隔的整数倍（默认会微调网格间隔以适应给定的数据范围）

- + 表明 <xinc> 和 <yinc> 不是网格间隔，而是 X 和 Y 方向的节点数

## 示例

```
gmt grdraster 1 -R-4/364/-62/62 -I30m -Gdata.nc
gmt grdraster ETOP02 -R160/20/220/30r -Joc190/25.5/292/69/1 -Gdata.nc
gmt grdraster "2 min Geoware" -R20/25/-10/5 -bo > triplets.b
```

---

**grdredpol**

---

官方文档 [grdredpol](#)

---

## grdconvert

---

官方文档 [grdconvert](#)

简介 在不同的网格格式之间互相转换

### 选项

<ingrdfile> 要读入的网格文件，文件名格式见 [grid-data](#)

<outgrdfile> 要写入的网格文件，文件名格式见 [grid-data](#)

-N 不将 GMT 头段结构写到文件中。该选项用于写一个无头段的 native 的二进制文件，该文件可直接用于 [grdraster](#) 命令中。

### 选项

从一个三维网格文件中提取第二层数据:

```
gmt grdconvert climate.nc?temp[1] temp.nc -V
```

将网格文件转换成四字节 native 浮点型网格:

```
gmt grdconvert data.nc ras_data.b4=bf -V
```

将网格文件转换成二字节短整型文件，将其乘以 10 并减去 32000，并设置无数据节点的值为-9999:

```
gmt grdconvert values.nc shorts.i2=bs/10/-32000/-9999 -V
```

将网格文件转换为 [grdraster](#) 可直接使用的二进制文件:

```
gmt grdconvert etopo2.nc etopo2.i2=bs -N -V
```

To creat a dumb file saved as a 32 bits float GeoTiff using GDAL:

```
gmt grdmath -Rd -I10 X Y MUL = lixo.tiff=gd:GTiff
```

---

**grdrotater**

---

官方文档 [grdrotater](#)



## grdsample

官方文档 [grdsample](#)

**简介** 对网格文件做重采样

该命令读取一个网格文件，并对其做插值以生成一个新的网格文件。新旧网格文件的区别在于：

1. 不同的配准方式（`-r` 或 `-T`）
2. 不同的网格间隔或网格节点数（`-I`）
3. 不同的网格范围（`-R`）

## 选项

`<in_grdfile>` 要重采样的 2D 网格文件

`-G<out_grdfile>` 重采样生成的网格文件

`-I<xinc>[<unit>][=|+]/<yinc>[<unit>][=|+]` 指定 X 和 Y 方向的网格间隔，见 [xyz2grd](#) 中的介绍。

`-T` 交换网格文件的配准方式。即若输入是网格线配准，则输出为像素点配准；若输入是像素点配准，则输出为网格线配准。

`-n[b|c|l|n][+a][+b<BC>][+c][+t<threshold>]` 重采样时使用的插值算法，见 [option-n](#) 一节。

## 示例

将 5x5 弧分的数据采样成 1x1 弧分：

```
gmt grdsample hawaii_5by5_topo.nc -I1m -Ghawaii_1by1_topo.nc
```

将网格线配准的网格文件修改为像素配准的网格文件：

```
gmt grdsample surface.nc -T -Gpixel.nc
```

---

**grdseamount**

---

官方文档 [grdseamount](#)

---

**grdspotter**

---

官方文档 [grdspotter](#)

---

## grdtrack

---

官方文档 [grdtrack](#)

**简介** 沿着指定的 (x,y) 位置采样网格点

该命令会读取一个或多个网格文件以及一个表数据。表数据中前两列分别是经纬度，其余列可忽略。该命令会提取指定位置处网格的值，若有必要，会对网格进行插值。

## 选项

**-G<grdfile>** 要采样的 2D 网格文件，若需要同时采样多个网格文件，可以重复使用 -G 选项。

**<xyfile>** 表数据，前两列为 (x,y) 位置信息

**-Z** 仅输出采样后的 Z 值，默认会输出所有列数据。

## grdtrend

官方文档 [grdtrend](#)

**简介** 拟合网格的趋势面并计算残差

该命令会读取一个 2D 网格文件，并用最小二乘方法拟合一个低阶多项式趋势面。多项式趋势面的定义为：

$$m1 + m2*x + m3*y + m4*x*y + m5*x*x + m6*y*y + m7*x*x*x + m8*x*x*y + m9*x*y*y + m10*y*y*y.$$

## 选项

**<gridfile>** 2D 网格文件名

**-N<n\_model>[r]** 指定要拟合的模型。

**<n\_model>** 指定要拟合的模型的参数个数。例如 **-N3** 表示 bilinear 趋势，**-N6** 表示 quadratic 趋势面。加上 **r** 表示 robust 拟合，此时，程序会根据 robust scale estimate 多次迭代，给数据重新赋予权重，以得到一个对 outliers 不敏感的解。

**-D<diff.nc>** 将残差（输入减去拟合）结果写到网格文件中

**-T<trend.nc>** 将拟合得到的趋势文件写到网格文件 **<trend.nc>** 中

**-W<weight.nc>** 若 **<weight.nc>** 存在，则读取该文件，并求解一个有权重的最小二乘问题。默认为常规的最小二乘拟合。若 **-N** 选项中指定了 robust 拟合，则 robust 拟合中所使用的权重会写到文件 **<weight.nc>** 中。

## 示例

从网格文件中移除线性趋势，并将结果写到残差文件中：

```
gmt grdtrend hawaii_topo.nc -N3 -Dhawaii_residual.nc
```

对网格文件做 bicubic 面的 robust 拟合:

```
gmt grdtrend hawaii_topo.nc -N10r -Thawaii_trend.nc -Whawaii_weight.nc -V
```

---

**grdvector**


---

**官方文档** [grdvector](#)**简介** 根据两个网格文件绘制矢量场

该命令会读取两个 2D 网格文件并绘制矢量场。两个网格文件分别代表平面内矢量场的 X 分量和 Y 分量，最终矢量场用不同长度和方向的箭头表示。两个网格文件，也可以分别是极坐标下  $r$  方向和  $\theta$  方向的分量。

**选项****<comp $x$ .nc>** 矢量场的 X 分量网格**<comp $y$ .nc>** 矢量场的 Y 分量网格**-A** 输入数据为极坐标表示。即网格文件包含的是 ( $r$ ,  $\theta$ ) 分量而不是 ( $x$ ,  $y$ ) 分量**-C[<cpt>]** 根据矢量的长度决定矢量的颜色。

有三种方式：

1. 指定用户自己的 CPT 文件
2. 使用 GMT 自带的 CPT 文件（默认是 rainbow），该命令会根据网格的 Z 值范围，自动生成一个 16 级的连续 CPT 文件
3. 指定 **-C<color1>,<color2>[,<color3>,...]**，根据这些颜色自动构建一个线性连续 CPT 文件

**-G<fill>** 设置矢量内部的填充色**-I[x]<dx>[/<dy>]** 每隔  $\langle dx \rangle$  和  $\langle dy \rangle$  绘制一个矢量。其中  $\langle dx \rangle$  和  $\langle dy \rangle$  必须是原始网格间隔的整数倍， $\langle dx \rangle$  和  $\langle dy \rangle$  后加上 m 表示弧分，加上 s 表示弧秒。也可以使用 **-Ix<dx>/<dy>**，此时  $\langle dx \rangle$  和  $\langle dy \rangle$  表示倍数。**-N** 不裁剪地图边界外的矢量。



**-Q<parameters>** 修改矢量的属性。见 [vectors](#) 一节。

**-S[i|l]<scale>** 设置矢量长度的缩放比例。

对于笛卡尔数据而言，缩放比例即图上单位距离所对应的实际数据的多少，默认值为 1。可以加上 `clip` 以指定测量单位。

**-Sl<scale>** 表示所有矢量拥有固定的长度。

对于地理数据而言，缩放比例表示每千米所对应的数据单位。使用 **-Si<scale>** 则缩放比例表示每数据单位所对应的千米数。

**-T** 该选项意味着笛卡尔数据的方位角会根据 X 和 Y 方向缩放比例的符号而改变。

**-W<pen>** 设置矢量轮廓的画笔属性。

**-Z** 与 **-A** 选项一起使用，表明输入的 `theta` 分量数据表示方位角而不是方向信息。

## 示例

对矢量长度进行缩放，使得图上 1 英寸代表实际的 10 个数据单位：

```
gmt grdvector r.nc theta.nc -Jx5c -A -Q0.1i+e+jc -S10i > gradient.ps
```

绘制地理数据，缩放比例为每个数据单位对应 200 km，每隔 3 个网格点绘制一个矢量：

```
gmt grdvector comp_x.nc comp_y.nc -Ix3 -JH0/20c -Q0.1i+e+jc -S200 > globe.ps
```

---

**grdview**

---

官方文档 [grdview](#)

---

## grdvolume

---

### 官方文档 [grdvolume](#)

**简介** 计算网格数据中某个等值线所包围的表面积和体积

该命令读取一个 2D 网格文件，通过指定某条等值线确定某个 Z 值平面，并计算由该等值线约束的区域网格表面积、网格表面到该平面所包围的体积，以及最大平均高度（体积/面积）。也可以指定一系列等值线，此时该命令会分别计算每个等值线范围内的表面积和体积。

## 选项

**<grdfile>** 输入的 2D 网格文件名

**-C** 指定等值线（Z 值平面）。

该选项有三种不同的语法：

- **-C<cval>** 指定单个等值线
- **-C<low>/<high>/<delta>** 指定多条等间距的等值线
- **-Cr<low>/<high>** 计算两个 Z 值平面之间的体积

**-L<base>** 同时也计算从等值线到 **<base>** 的体积

**-S[<unit>]** 将弧度转换为 Flat Earth 距离，默认为笛卡尔坐标

**-T[c|h]**

- **-Th** 确定最大平均高度所对应的等值线
- **-Tc** 确定最大曲率（高度 vs 等值线值）所对应的等值线

**-Z<fact>[/<shift>]** 将数据减去 **<shift>** 再乘以比例因子 **<fact>** 。

## 示例

计算网格表面之下的体积，网格数据的单位是 km，结果的单位为  $\text{km}^3$ :

```
gmt grdvolume hawaii_topo.nc -Sk
```

计算网格表明与等值线  $Z=250$  之间的体积:

```
gmt grdvolume peaks.nc -Sk -C250
```

在等值线 100 到 300 范围内，以 10 为间隔，计算所有等值线所约束的表面积和体积:

```
gmt grdvolume peaks.nc -Sk -C100/300/10 > results.d
```

在等值线 100 到 300 范围内，以 10 为间隔，搜索最大平均高度（即体积与表面积之比）所对应的等值线值:

```
gmt grdvolume peaks.nc -Sk -C100/300/10 -Th > results.d
```

计算湖内从表面到 300 米深度范围内水的体积:

```
gmt grdvolume lake.nc -Cr-300/0
```

---

**greenspline**

---

官方文档 [greenspline](#)

---

**gshhg**

---

官方文档 [gshhg](#)

---

**hotspotter**

---

官方文档 [hotspotter](#)

---

**img2google**

---

官方文档 [img2google](#)



---

**img2grd**

---

官方文档 [img2grd](#)

---

## isogmt

---

官方文档 [isogmt](#)

**简介** 在“隔离模式”下运行 GMT 命令或脚本

该命令本质上是一个 Bash 脚本，其脚本内容为：

```
export GMT_TMPDIR=`mktemp -d ${TMPDIR:-/tmp}/gmt.XXXXXX`  
gmt "$@"  
rm -rf $GMT_TMPDIR  
unset GMT_TMPDIR
```

该脚本首先定义了环境变量 `GMT_TMPDIR`，用于给 GMT 指定临时目录，接下来执行的 GMT 命令所生成的临时文件都会保存在该临时目录下。待命令/脚本执行完毕后，再删除临时目录和环境变量。

## 语法

```
isogmt command
```

## 示例

隔离模式下运行一个 GMT 命令：

```
isogmt psbasemap -R0/10/0/10 -JX10c/10c -B1 > test.ps
```

隔离模式下运行一个脚本（bug）：

```
isogmt sh run.sh
```

## BUGS

1. 无法在隔离模式下执行脚本 (v5.2.1)

---

## kml2gmt

---

官方文档 [kml2gmt](#)

**简介** 将 Google Earth 的 KML 文件转换为 GMT 表数据

只有包含点、线或多边形的 Google Earth KML 才能被转换。

### 选项

**<kmlfiles>** 要转换的 KML 文件

**-F[s|l|p]** 指定要输出的特征类型

默认会输出 KML 中所包含的所有点、线或多边形

1. **-Fs** 只输出点
2. **-Fl** 只输出线
3. **-Fp** 只输出多边形

**-Z** 默认只输出经纬度信息，若使用 **-Z** 选项，则输出坐标的高程信息作为 GMT 的 Z 值

### 示例

```
gmt kml2gmt google.kml -V > google.txt
```

从一个 KML 文件中分别提取点和多边形到不同的文件:

```
gmt kml2gmt google.kml -Fp -V > polygons.txt
gmt kml2gmt google.kml -Fs -V > points.txt
```

也可以直接用外部命令 `ogr2ogr` 实现转换:

```
ogr2ogr -f "GMT" somefile.gmt somefile.kml
```

---

**makecpt**

---

官方文档 [makecpt](#)

简介 制作 GMT CPT 文件

## 选项

**-A[+]<transparency>** 为所有的颜色切片设置相同的透明度，<transparency> 的取值范围是 0 到 100。  
加上 + 则该透明度同时应用于前景色、背景色和 NaN 颜色。

**-C<cpt>** 指定要进行插值的主 CPT 文件，默认值是 `rainbow` 。

可以是 GMT 自带的 CPT 文件，也可以是用户自定义的 CPT 文件，也可以是通过  
**-C<color1>,<color2>[,<color3>]** 语法指定一个线性连续变化的颜色列表。

**-T<z\_min>/<z\_max>[/<z\_inc>[+]]** 或 **-T<ztable>** 定义要生成的 CPT 文件的 Z 值范围及 Z 值间隔

- 若使用了 **-C** 选项且 <z\_inc> 未指定，则 Z 值间隔的数目与输入的主 CPT 文件相同。
- 若 <z\_inc> 后有 + ，则将 <z\_inc> 解释为 Z 值间隔的数目而不是 Z 值间隔
- 也可以指定文件 <ztable> ，其中每行一个 Z 值
- 若不使用 **-T** 选项，则默认使用主 CPT 文件中的 Z 值范围

**-Z** 生成连续 CPT 文件。默认生成不连续 CPT 文件，即每个 Z 值切片内为同一颜色

---

## mapproject

---

官方文档 [mapproject](#)

- 投影正变换：读入经纬度数据，将其转换为指定投影类型下的 XY 坐标
- 投影逆变换：读入 XY 坐标，将其转换为经纬度数据

### 最小示例

最简单的，mapproject 可以用于将计算任意一点相对于当前底图左下角的偏移量：

```
$ echo 5 5 | mapproject -R0/10/0/10 -JX10c/5c
5 2.5
```

-JX10c/5c 指定了投影方式，-R0/10/0/10 指定了数据范围，心算一下可知坐标 (5,5) 相对于底图左下角的偏移距是 (5c,2.5c)，也就是 mapproject 的输出。

对于地理投影，投影公式很复杂，心算就不可能了。mapproject 的用处就体现出来了：

```
$ echo 121 42 | mapproject -R100/160/0/80 -JM10c
3.5 7.6841597874
```

由输出可知，坐标 (121,42) 相对于当前底图原点的偏移距是 (3.5c,7.68c)。

### -I 选项

-I 选项表明做投影逆变换，即从 XY 坐标中得到经纬度信息：

```
$ echo 3.5 7.6841597874 | mapproject -R100/160/0/80 -JM10c
121 42
```

## **-Q 选项**

- -Qd : 列出 GMT 支持的全部大地水准面模型
- -Qe : 列出 GMT 支持的全部椭球模型
- -Q : 列出 GMT 支持的全部椭球模型和大地水准面模型



---

**mgd77convert**

---

官方文档 [mgd77convert](#)

---

**mgd77info**

---

官方文档 [mgd77info](http://mgd77info)

---

**mgd77list**

---

官方文档 [mgd77list](#)

---

**mgd77magref**

---

官方文档 [mgd77magref](#)

---

**mgd77manage**

---

官方文档 [mgd77manage](#)

---

**mgd77path**

---

官方文档 [mgd77path](#)

---

**mgd77sniffer**

---

官方文档 [mgd77sniffer](#)

---

**mgd77track**

---

官方文档 [mgd77track](#)



---

**nearneighbor**

---

官方文档 [nearneighbor](#)

---

**originator**

---

官方文档 [originator](#)

## project

官方文档 [project](#)

**简介** 将数据点投影到线或大圆路径上，生成测线，坐标转换

不管是要生成测线还是要将数据投影到测线上，首先要定义测线。

测线可以用如下三种方式中的任意一种来定义：

1. 用 `-C` 选项定义测线的起点，并用 `-A` 定义测线方位角
2. 用 `-C` 选项定义测线的起点，并用 `-E` 选项定义测线的结束点
3. 用 `-C` 选项定义测线的中心，并用 `-T` 选项定义 rotation pole 的位置

在指定测线后，该命令可以将输入的 (x, y) 数据投影到该测线上，默认的输出格式为：

```
x y z p q r s
```

其中：

- `x` 和 `y` 是数据在原坐标系下的位置，
- `z` 是输入数据中的其余所有列
- `p` 和 `q` 是数据 (x,y) 在测线坐标系下的位置
- `r` 和 `s` 是测线上与 (x,y) 最近的点在 XY 坐标系下的坐标

关于测线坐标系，在指定了测线后，会创建一个倾斜的坐标系。该倾斜坐标系的赤道线沿着测线，零经度线则穿过测线的中心，此时 `p` 是倾斜坐标系下的经度，`q` 是倾斜坐标系下的纬度。

## 选项

`-C<cx>/<cy>` 对于定义 1 和 2 而言，该选项指定测线的起点坐标；对于定义 3 而言，该选项指定了旋转坐标下零经线所穿过的点

`-A<azimuth>` 定义 1 中用于指定测线的方位角

**-E<bx>/<by>** 定义 2 中用于指定测线的终点

**-T<px>/<py>** 定义 3 中用于指定 rotation pole 的位置

**-F<flags>** 指定输出格式，flags 可以是 xyzpqrs 的任意组合。

默认输出格式为 xyzpqrs 。若使用 -G 选项，则输出格式为 rsp

**-G<dist>[/<colat>]** 生成测线模式。

该选项用于生成测线，此时不需要输入文件。输出数据有三列：经度、纬度以及当前点离测线起点的距离。默认情况下，会按照大圆路径生成测线，当使用定义 2 指定测线的两个端点时，可以通过指定 <colat> 来生成小圆。

**-L[w][<l\_min>/<l\_max>]** 仅坐标 p 在 <l\_min> 和 <l\_max> 之间的数据会被投影到测线上。若使用了 -E 选项，则可以使用 -Lw 来限制只绘制能投影到测线起点和终点之间的数据。

**-N** Flat Earth。在平面内使用笛卡尔坐标变换。默认使用球面三角。

**-Q** 使用地图单位。即投影假设 x、y、r、s 的单位是弧度，p、q、<dist> 、<l\_min> 、<l\_max> 、<w\_min> 、<w\_max> 的单位是千米。若不使用该选项，则所有量都假设为相同的单位。

**-S** 将输出按照 p 增序排列。

**-W<w\_min>/<w\_max>** 仅坐标 q 在 <w\_min> 和 <w\_max> 范围内的数据才会被投影到测线上。

## 示例

指定测线的起点和终点，在测线上每隔 10 千米生成一个点：

```
gmt project -C-50/10 -E-10/30 -G10 -Q > great_circle_points.xyp
```

指定测线的起点和终点，沿着 colatitude=60 的小圆上，每隔 10 千米生成一个点：

```
gmt project -C-50/10 -E-10/30 -G10/60 -Q > small_circle_points.xyp
```

---

## psbasemap

---

官方文档 [psbasemap](#)

简介 绘制底图

该命令用于绘制底图边框（标注、刻度、标签等）、标题、比例尺、方向玫瑰等。

## 选项

**-B** 绘制底图边框，见关于 **-B** 选项的介绍。

示例代码：

```
gmt psbasemap -R-180/180/-70/70 -JM10c -Bx60 -By30 > test.ps
```

**-A[<file>]** 不绘制图形，仅输出矩形底图的边框坐标。

该选项会将矩形底图的边框坐标输出到标准输出或文件中。使用该选项时，必须通过 **-J** 和 **-R** 指定绘图区域，且不能同时使用其他选项。若不指定 **<file>** 则默认输出到标准输出，否则输出到文件 **<file>** 中

说明：

1. 该选项似乎仅适用于矩形底图边框，非矩形边框会输出一堆 NaN
2. 尚不清楚该选项存在的意义
3. 边框的采样间隔由参数 **MAP\_LINE\_STEP** 决定
4. 该选项似乎有 bug

示例：

```
gmt psbasemap -R0/10/0/10 -JX10c/5c -Aoutline.txt
```

**-D** 在底图中绘制图中图（insert map box），需要 **-F** 选项，其语法有两种：

```
-D[<unit>]<xmin>/<xmax>/<ymin>/<ymax>[r][+s<file>]
-D[g|j|J|n|x]<refpoint>+w<width>[/<height>][+j<justify>][+o<dx>[/<dy>]][+s<file>]
```

先解释图中图 (insert box) 的概念。在绘制一个小区域时，为了表明该小区域在地球上的位置，通常需要在一张更大范围（比如整个中国或整个地球）的底图上标出小区域所在的位置，这种绘图形式通常称为 insert map。而 insert box 就是 insert map 的边框。

指定 insert map box 的范围有三种方法：

1. `-D<xmin>/<xmax>/<ymin>/<ymax>[r]`：类似 `-R` 的语法，直接指定 insert box 在地图上的范围
2. `-D<unit><xmin>/<xmax>/<ymin>/<ymax>`：类似 `-R` 的语法，指定投影后的坐标范围，`<unit>` 为投影后的坐标单位
3. 指定参考点和锚点，见 [embellishments](#) 一节
  - (a) `[g|j|J|n|x]<refpoint>` 指定大区域底图上的参考点
  - (b) `+j<justify>` 指定 insert box 上的锚点
  - (c) `+o<dx>/<dy>` 指定参考点的额外偏移量
  - (d) `+w<width>[/<height>]` 指定 insert box 的宽度或/和高度

使用 `+s<file>` 选项，会将 insert box 的左下角位置以及其长宽写到文件中，坐标值以及长度值均使用当前地图单位。

**-F** 控制 insert box、比例尺和方向玫瑰的背景边框的属性。

其语法为：

```
-F[d|l|t][+c<clearances>][+g<fill>][+i[[<gap>/]<pen>]][+p<pen>][+r<radius>][+s[[<dx>/<dy>/<→>][<shade>]]]
```

说明：

1. 见 [embellishments](#) 一节的详细介绍
2. 该选项用于给 insert box、比例尺和方向玫瑰绘制背景边框，默认同时控制三者的属性
3. `d|l|t` 表示该选项定义的属性仅适用于 `-D`、`-L` 或 `-T` 选项
4. 直接使用 `-F` 选项，则绘制背景边框，边框属性由参数 `MAP_FRAME_PEN` 控制

**-L** 在地图上绘制比例尺。

该选项的语法为：

```
-L[g|j|J|n|x]<refpoint>+c[<slon>/]<slat>+w<length>[e|f|k|M|n|u][+a<align>][+f][+l<label><→>][+u]
```

说明：

1. 大部分在 [embellishments](#) 一节已经介绍了

2. `[g|j|J|n|x]<refpoint>` 指定地图上的参考点，比例尺的锚点位于比例尺的中心
3. `+c<slon>/<slat>` 要绘制哪一个点的比例尺
4. `+w<length>[e|f|M|n|k|u]` 比例尺的长度，默认单位为 km，也可使用其他长度单位
5. `+a<align>` 修改比例尺标签的位置，默认位于比例尺上方中文，可以取 `l`、`r`、`t``、`b` 分别代表左右上下
6. `+l<label>` 为比例尺加标签；若不指定 `<label>`，默认的标签是比例尺所使用的长度单位
7. `+u` 比例尺的标注默认只有值没有单位，该选项会给标注加上单位
8. `+f` 默认是简单的比例尺，使用该选项则绘制 fancy 比例尺，即火车轨道比例尺

相关参数：

1. `FONT_LABEL` 控制比例尺的标签字体
2. `FONT_ANNOT_PRIMARY` 控制比例尺的标注字体
3. `MAP_SCALE_HEIGHT` 控制比例尺的高度
4. `MAP_TICK_PEN_PRIMARY` 控制比例尺的刻度属性

示例：

```
gmt psbasemap -R90/110/30/40 -JM10c -Bx5 -By5 -Lg95/35+c35+w800k+lscale+u+f > test.ps

gmt psbasemap -R90/180/-50/50 -Jm0.025i -Bafg -B+tMercator -Lx1i/1i+c0+w5000k > mercator.ps
```

**-Td** 绘制方向玫瑰图，其语法为：

```
-Td[g|j|J|n|x]<refpoint>+w<width>[+f[<level>]] [+j<justify>] [+l<w,e,s,n>] [+o<dx>[/<dy>]]
```

说明：

1. 大部分在 [embellishments](#) 一节已经介绍了
2. `[g|j|J|n|x]<refpoint>` 指定地图上的参考点
3. `+j<justify>` 指定玫瑰图上的锚点（默认为 MC）
4. `+o<dx>/<dy>` 指定参考点的偏移量
5. `+w<width>` 玫瑰图的宽度
6. `+f<level>` 绘制 fancy 玫瑰图，`<level>` 指定了 fancy 玫瑰图的不同类型，`<level>` 可以取：
  - 1 绘制 E-W 和 N-S 四个方向
  - 2 绘制 8 个方向
  - 3 绘制 16 个方向

7. `+l<w>,<e>,<s>,<n>` 为四个方向分别指定标签，默认标签是四个方向的单字母代码（英文语言下是 W、E、S、N，具体值由参数 `GMT_LANGUAGE` ），四个方向的标签之间用逗号分隔，比如 `+lw,e,s,n` 或 `+l",,Down,Up"`

下图展示了方向玫瑰图的效果图：

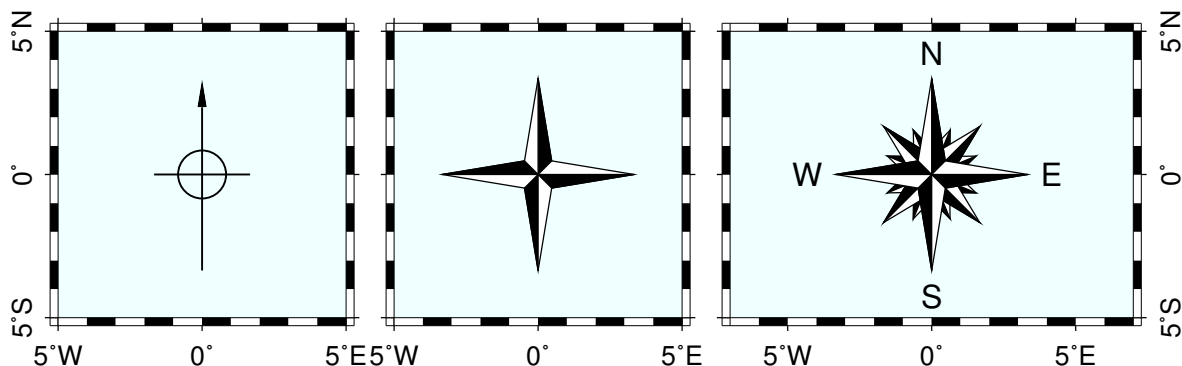


图 83.1: 方向玫瑰图

(左) `-Tdg0/0+w1i+jCM` (中) `-Tdg0/0+w1i+f1+jCM` (右) `-Tdg0/0+w1i+f3+l+jCM`

**-Tm** 绘制磁场玫瑰图，用于展示磁场方向。

其语法为：

```
-Tm[g|j|J|n|x]<refpoint>+w<width>[+d<dec>[/<dlabel>]] [+i<pen>] [+j<justify>] [+l<w>,<e>,<s>,<n>]
→] [+p<pen>] [+t<ints>] [+o<dx>[/<dy>]]
```

磁场玫瑰包括两个同心圆环，其中外环用于展示方向信息，内环用于显示磁场方向。

说明：

1. 大部分在 [embellishments](#) 一节已经介绍了
2. `[g|j|J|n|x]<refpoint>` 指定地图上的参考点
3. `+j<justify>` 指定玫瑰图上的锚点（默认为 MC ）
4. `+o<dx>[/<dy>]` 指定参考点的偏移量
5. `+w<width>` 玫瑰图的宽度
6. `+p<pen>` 绘制外环的轮廓
7. `+i<pen>` 绘制内环的轮廓
8. `+d<dec>[/<dlabel>]` 设置磁倾角以及罗盘指针上的磁倾角标签。若 `<dlabel>` 为空，则使用默认标签 `d = <dec>` ；若 `<dlabel>` 为 `-` ，则不绘制标签。当使用 `+d` 子选项时，会同时绘制地理方向和磁场方向
9. `+l<w>,<e>,<s>,<n>` 为四个方向的标签，默认标签是四个方向的单字母代码（英文语言下是 W、E、S、N，具体值由参数 `GMT_LANGUAGE` ；若 `<n>` 取值为 `*` ，则会在北方向标签处绘制星代表北极星



10. 内外环都可以设置标注、刻度和网格的间隔。内外环的间隔默认值都是 30/5/1 。可以使用 `+t<ints>` 选项，后面接 6 个斜杠分隔的值，以分别指定两个圆环的 3 种刻度值，其中前三个值控制内环属性，后三个值控制外环属性

## 示例

下面的脚本展示了一种 insert box 的用法：

```
#!/bin/bash
R=115/123/35/40          # 小区域地图的范围 -R
J=M20c                  # 小区域的投影方式 -J
PS=psbasemap_ex1.ps

# 打开 GMT
gmt psxy -R$R -J$J -T -K > $PS

# 绘制小区域的地图
gmt pscoast -R$R -J$J -B1 -Df -W2p -I2p -N1/2p -A1000 -K -O >> $PS
# ...
# 此处可以任意添加其他绘图命令
# ...

# 开始绘制 insert map
Rg=72/135/16/55         # 大区域地图的范围 -R
Jg=M7c                  # 大区域地图的投影方式 -J
# 绘制大区域地图的海岸线及边框
gmt pscoast -R$Rg -J$Jg -B0 -B+gwhite -Df -N1 -W -A5000 -K -O --MAP_FRAME_TYPE=plain >> $PS
# 在大区域地图内绘制小区域所在的方框
gmt psbasemap -R$Rg -J$Jg -D$R -F+p2p,blue -K -O >> $PS

# 结束 GMT 绘图
gmt psxy -R$R -J$J -T -O >> $PS
```

下面的脚本绘制了另一种不同的 insert box：

```
#!/bin/bash
ps=psbasemap_ex2.ps

# 绘制澳大利亚的海岸线
gmt pscoast -R110E/170E/44S/9S -JM6i -P -Baf -BWSne -Wfaint -N2/1p -EAU+gbisque \
    -Gbrown -Sazure1 -Da -K -Xc --FORMAT_GEO_MAP=dddF > $ps

# 在右上角绘制 insert box
gmt psbasemap -R -J -O -K -DjTR+w1.5i+o0.15i+stmp -F+gwhite+p1p+c0.1c+s >> $ps
```

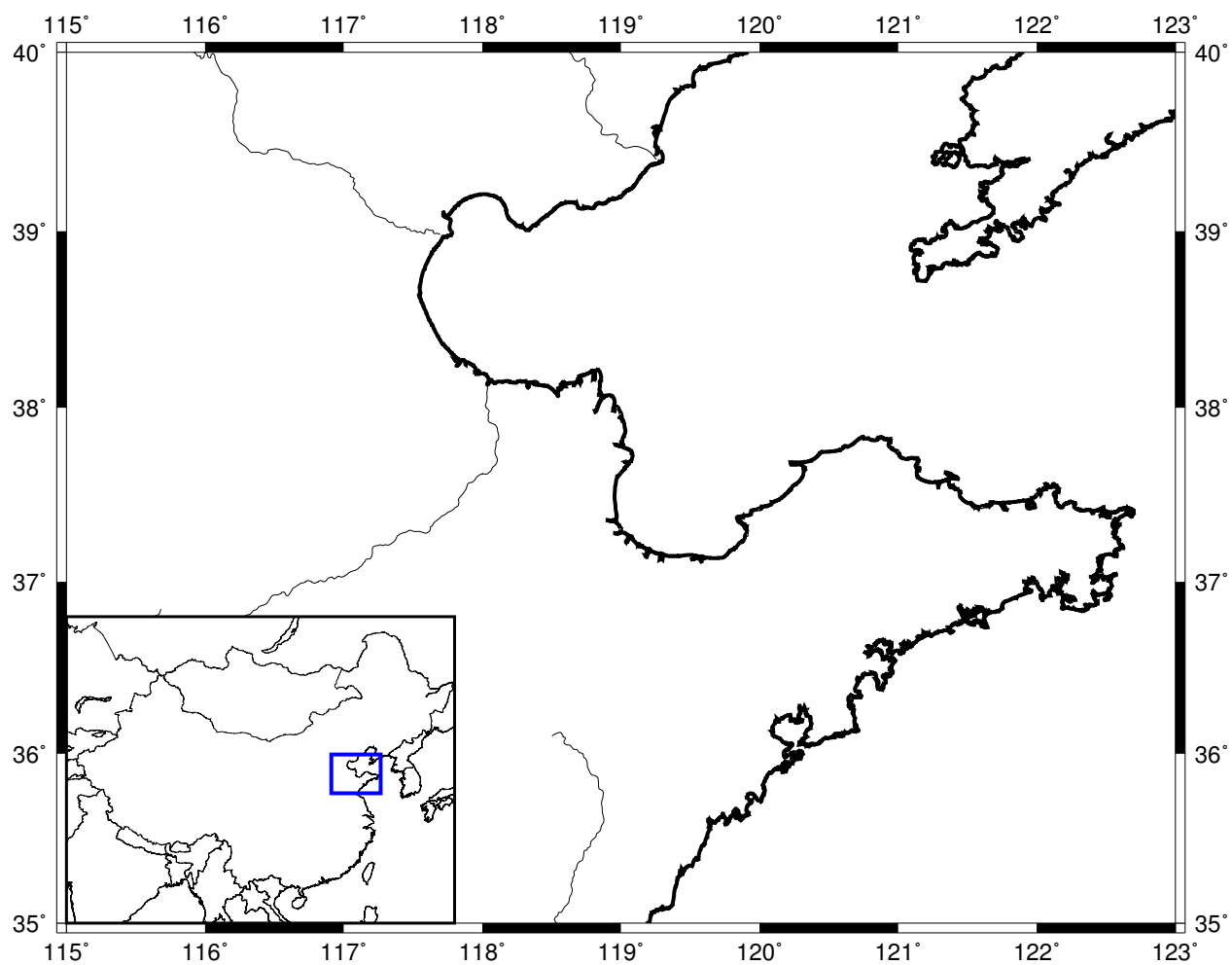


图 83.2: 用 psbasemap 命令-D 选项绘制图中图

```

# 从临时文件中读取 insert box 的左下角的位置
read x0 y0 w h < tmp
# 在 insert box 内绘制全球地图，并突出显示澳大利亚
gmt pscoast -Rg -JG120/30S/$w -Da -Gbrown -A5000 -Bg -Wfaint -EAU+gbisque -O -K -X$x0 -Y$y0 >> $ps

gmt psxy -R -J -O -T -X-$x0 -Y-$y0 >> $ps

rm gmt.* tmp

```

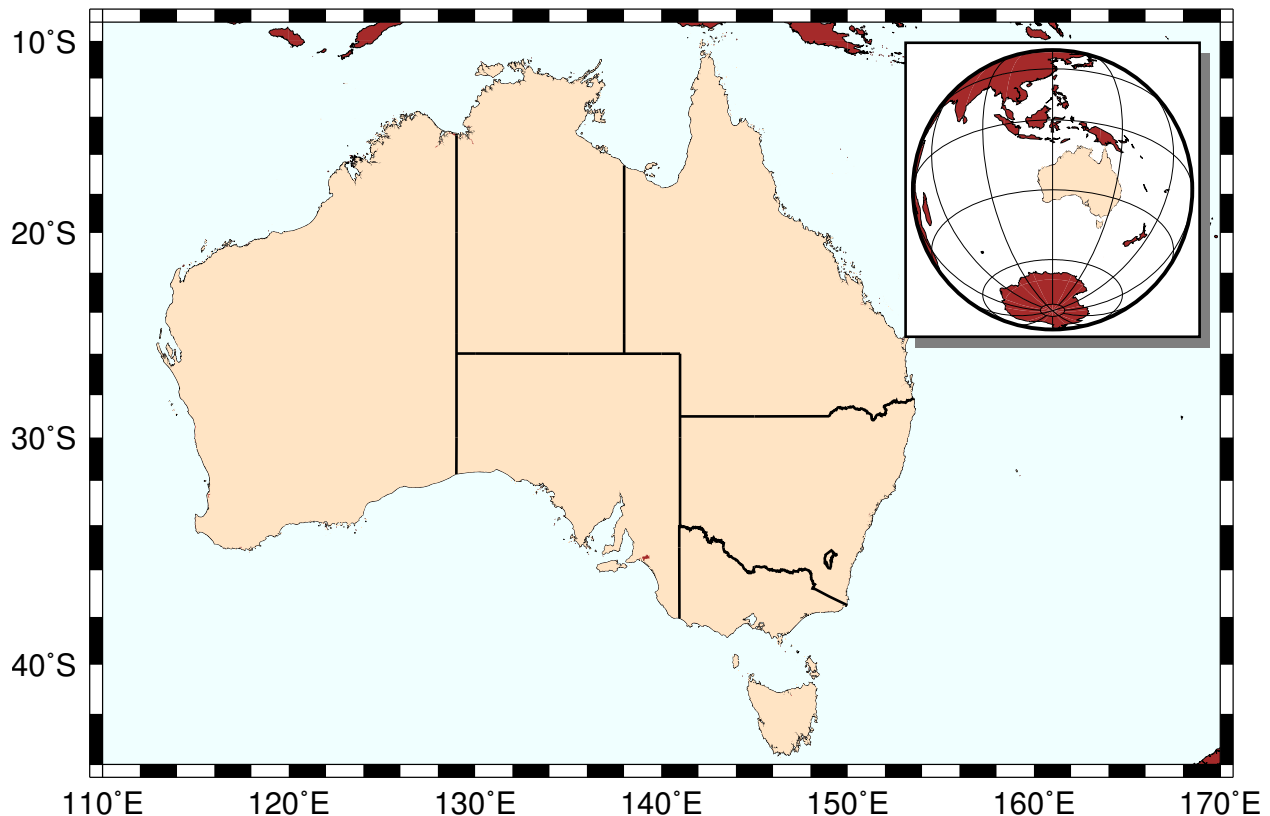


图 83.3: 用 psbasemap 命令-D 选项绘制另一种图中图

下面的脚本绘制了磁场玫瑰图：

```

#!/bin/bash
# $Id$
ps=psbasemap_ex4.ps
# Magnetic rose with a specified declination
gmt psbasemap -R-10/-2/12.8812380332/0.661018975345r -J0c0/0/50/60/7i -Baf -BWSne -P -K -X1.25i \
--MAP_ANNOT_OBLIQUE=34 --FONT_ANNOT_PRIMARY=12p > $ps
gmt psbasemap -R -J -Tmg-2/0.5+w2.5i+d-14.5+t45/10/5+i0.25p,blue+p0.25p,red+l+jCM -O -K \
--FONT_ANNOT_PRIMARY=9p,Helvetica,blue --FONT_ANNOT_SECONDARY=12p,Helvetica,red \
--FONT_LABEL=14p,Times-Italic,darkgreen --FONT_TITLE=24p --MAP_TITLE_OFFSET=7p \
--MAP_FRAME_WIDTH=10p --COLOR_BACKGROUND=green --MAP_DEFAULT_PEN=2p,darkgreen \

```

```

--COLOR_BACKGROUND=darkgreen --MAP_VECTOR_SHAPE=0.5 --MAP_TICK_PEN_SECONDARY=thinner,red \
--MAP_TICK_PEN_PRIMARY=thinner,blue >> $ps
gmt psbasemap -R -J -O -K -DjTR+w2.9i/3.9i+o0.05i -F+p+ggray95 >> $ps
echo "5.5 3.8 GMT DEFAULTS" | gmt ptext -R0/7/0/5 -Jx1i -O -K -F+f14p,Helvetica-Bold+jCM >> $ps
gmt ptext -R -J -O -K -F+f12p+jLM << EOF >> $ps
4.1 3.50 FONT_TITLE
4.1 3.25 MAP_TITLE_OFFSET
4.1 3.00 MAP_DEGREE_SYMBOL
4.1 2.75 @;blue;FONT_ANNOT_PRIMARY@;;
4.1 2.50 @;blue;MAP_TICK_PEN_PRIMARY@;;
4.1 2.25 @;blue;MAP_ANNOT_OFFSET_PRIMARY@;;
4.1 2.00 @;blue;MAP_TICK_LENGTH_PRIMARY@;;
4.1 1.75 @;red;FONT_ANNOT_SECONDARY@;;
4.1 1.50 @;red;MAP_TICK_PEN_SECONDARY@;;
4.1 1.25 @;red;MAP_ANNOT_OFFSET_SECONDARY@;;
4.1 1.00 @;red;MAP_TICK_LENGTH_SECONDARY@;;
4.1 0.75 @;darkgreen;FONT_LABEL@;;
4.1 0.50 @;darkgreen;MAP_DEFAULT_PEN@;;
4.1 0.25 @;darkgreen;COLOR_BACKGROUND@;;
EOF
gmt psxy -R -J -O -T >> $ps
rm gmt.*

```

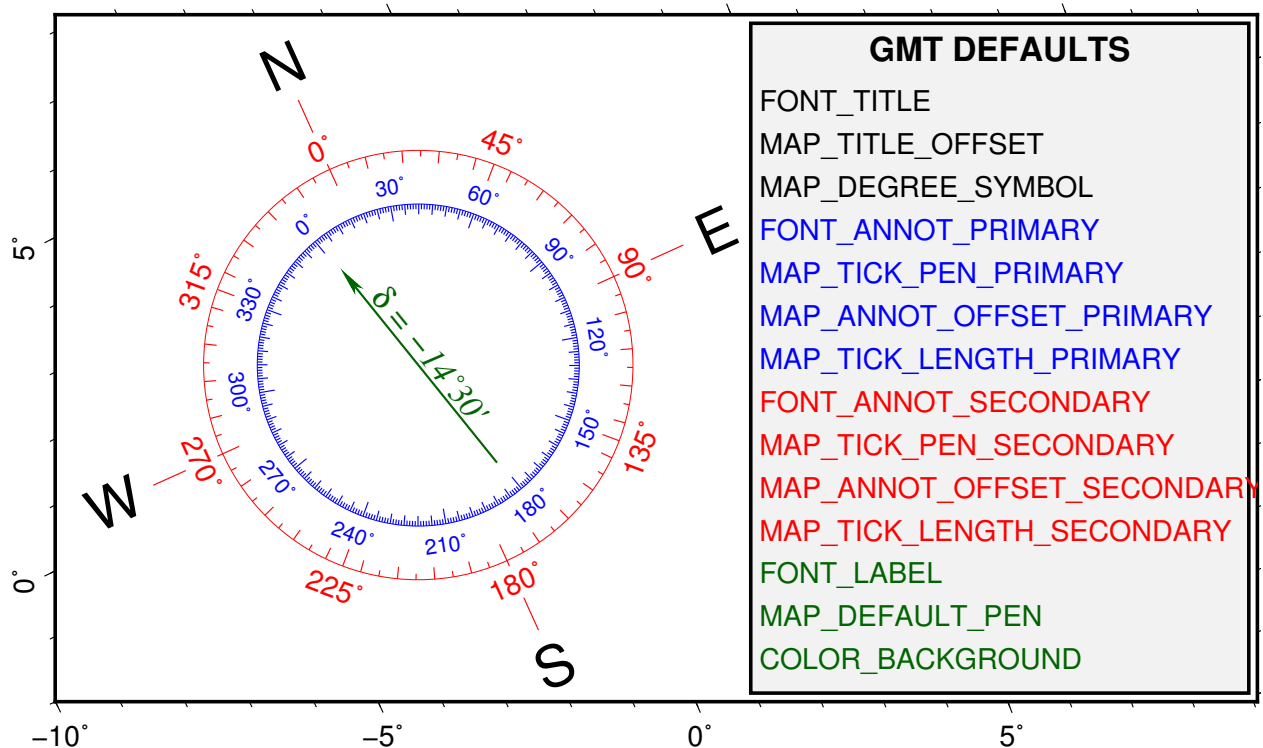


图 83.4: 磁场玫瑰图

## BUGS

1. 使用 `-A` 选项可以正常输出结果，但会出现 `double free` 的错误 (v5.2.1)

---

## psclip

---

### 官方文档 [psclip](#)

#### 简介 打开或关闭多边形裁剪路径

该命令会从输入文件中读取 XY 数据，由此构成一个或多个多边形，进而构建出一个或多个裁剪路径。接下来的所有绘图命令中，只有在多边形内部的部分才会被绘制。

为了判断某个点是在裁剪区域内还是在裁剪区域外，psclip 使用了奇偶规则。从任意一点绘制一条任意方向的射线，若该射线穿过裁剪路径线段奇数次，则该点位于裁剪区域内；若穿过偶数次，则该点位于裁剪区域外。

最后，记得再次调用 psclip 命令并使用 -C 选项以关闭裁剪区域。

## 选项

- A 修改两点间的连接方式，见 [psxy](#) 中 -A 选项的介绍
- C 结束当前裁剪路径。若在开启裁剪后有使用 -X|-Y 移动过坐标原点，记得要关闭裁剪路径时也有使用 -X|-Y
- N 反转在区域里还是在区域外的判断，进而导致只有在多边形外的部分才是裁剪区域，绘图时只有在多边形外的才会被绘制。
- T 不需要任何输入数据，仅仅按照当前地图的区域打开裁剪选项，该选项不能与 -B 选项连用。

## 示例

```
# 打开裁剪路径
gmt psclip my_region.xy -R0/40/0/40 -Jm0.3i -K > clip_mask_on.ps
# 其他绘图命令
# ...
```

```
# 关闭裁剪路径
```

```
gmt psclip -C -0 >> complex_plot.ps
```

## pscoast

官方文档 [pscoast](#)

**简介** 在地图上绘制海岸线、河流、国界线

该命令除了可以用于绘制海岸线、河流、政治边界，还可以裁剪陆地区域或水域，也可以将数据导出到文件中。

## 选项

**-A** 不绘制面积过小的湖泊的边界，或不绘制某个级别的湖泊边界。

该选项的语法是：

```
-A<min_area>[/<min_level>/<max_level>] [+ag|i|s|S] [+r|l] [+p<percent>]
```

在绘制湖泊时，若不管湖泊的面积大小而把所有湖泊的边界都画上去，可能导致图看上去比较乱，该选项用于对湖泊进行筛选。面积小于 `<min_area>` 平方千米或者湖泊级别不在 `[min_level,max_level]` 范围的边界都不会被绘制。默认值为 `0/0/4`，即绘制所有湖泊边界。关于湖泊边界的等级，在下面会介绍。

对于 `level=2`，即湖岸线，包括常规的湖以及很宽的河流。加上 `+r` 则只绘制河流，加上 `+l` 则只绘制常规湖。

对于南极洲而言，因为有冰层的存在，所以海岸线有多种处理方式：

1. `+ai` 用 ice shell boundary 作为南极洲的海岸线，默认值
2. `+ag` 以 ice grounding line 作为海岸线
3. `+as` 跳过南纬 60 度以南的海岸线，这样用户即可使用 `psxy` 绘制自己的南极洲海岸线
4. `+aS` 跳过南纬 60 度以北的海岸线

`+p<precent>`：一个多边形，降低精度后，边数减少，面积变化，当面积变化过大时再绘制这个多边形就不合适了，该子选项用于去除那些面积与最高精度面积之比小于 `<percent>` 的多边形。



**-C[l|r]<fill>** 设置湖泊与河流湖的颜色。

默认情况下，湖泊与河流湖会被当做 wet 区域，直接使用 -S 指定的填充值。该选项可以为湖泊和河流湖单独指定颜色，也可以多次使用该选项分别为湖泊和河流湖指定颜色。

1. **-G<fill>** 同时指定湖泊和河流湖的颜色
2. **-Gl<fill>** 指定湖泊 (lake) 颜色
3. **-Gr<fill>** 指定河流湖 (river-lake) 颜色

**-D[a|f|h|i|l|c][+]** 选择海岸线数据精度。

GMT 自带的 GSHHG 海岸线数据，有 5 个不同精度的版本，从高到低依次为：full、high、intermediate、low 和 crude。GMT 默认使用低精度数据。该选项可以指定要使用的数据精度，其中 **<resolution>** 可以取 f|h|i|l|c 分别代表 5 种不同的精度，当然也可以用 -Da 选项，此时 GMT 会根据当前绘图区域的大小自动选择合适的精度。

默认情况下，若找不到指定精度的海岸线数据，程序会自动报错退出。该选项中加上 + 则命令在找不到当前指定的精度数据时，自动寻找更低精度的数据。

**-E<code1>,<code2>,...[+l|L][+gfill][+ppen][+r|R[incs]]** 绘制或导出行政区划边界（洲界、国界、省界）。

除了海岸线数据 GSHHG 之外，GMT 还自带了 DCW (Digital Chart of World) 数据，即全球的行政区划数据。DCW 数据位于 `#{GMTHOME}/share/dcw` 目录下，包含了全球各国的国界和各国的省界数据。该数据独立于 GSHHG 数据，因而 -A 和 -D 选项对该数据无效。

说明：

1. **<code>** 是要绘制或提取的边界数据的代码，多个代码之间用逗号分隔
2. 具体的代码参考 dcw 目录下的文档。代码有如下几种形式
  - (a) 洲代码前加上 = 则绘制整个洲内所有国家边界，比如 =AS 会绘制所有亚洲国家的边界
  - (b) 直接使用国界代码，则绘制国界边界，比如 US 绘制美国边界
  - (c) 使用 国家代码. 州代码，则绘制州（省）边界，比如 US.TX 绘制美国 Texas 州的边界
3. **+l** 仅列出所有国家及其对应代码，不绘制边界也不提取数据
4. **+L** 列出部分国界的省及其代码
5. **+g<fill>** 设置多边形的填充色
6. **+p<pen>** 绘制多边形的轮廓
7. **+r** 获取多边形所对应的区域范围，以便于直接从数据中提取 -R 选项的范围。**<incs>** 可以是 **<inc>**、**<xinc>/<yinc>**、**<winc>/<einc>/<sinc>/<ninc>** 以调整区域范围使得范围是这些步长的整数倍
8. **+R** 与 **+r** 类似，只是之后的 **<incs>** 等参数被解释为区域范围向外扩展的增量

额外的说明：

1. 除非使用了 `+r` 、`+R` 或 `-M` 选项，否则必须指定 `+pl+g` 中的一个
2. `-E` 选项可以重复出现多次，以分别为不同的多边形设置不同的属性
3. 若使用了 `+r|R` 但未使用 `-J|-M` ，则直接输出 `-R<w>/<e>/<s>/<n>` 格式的字符串
4. 个别地方有 BUG

`-F` 控制比例尺和玫瑰图的背景边框，见 [embellishments](#) 一节

`-G<fill>|c` 设置 dry 区域的填充色或裁剪 dry 区域

1. `-G<fill>` 设置 dry 区域（一般指陆地）的填充色
2. `-Gc` 将 dry 区域裁剪出来，使得接下来的绘图只有 dry 区域内的才会被绘制

`-I<river>[/<pen>]` 绘制河流。

河流 `<river>` 可以取：

- 0 = Double-lined rivers (river-lakes)
- 1 = Permanent major rivers
- 2 = Additional major rivers
- 3 = Additional rivers
- 4 = Minor rivers
- 5 = Intermittent rivers - major
- 6 = Intermittent rivers - additional
- 7 = Intermittent rivers - minor
- 8 = Major canals
- 9 = Minor canals
- 10 = Irrigation canals
- a = All rivers and canals (0-10)
- A = All rivers and canals except river-lakes (1-10)
- r = All permanent rivers (0-4)
- R = All permanent rivers except river-lakes (1-4)
- i = All intermittent rivers (5-7)
- c = All canals (8-10)

`<pen>` 的默认值为 `default,black,solid` ，该选项可以重复使用多次。

`-L` 绘制比例尺。见 [psbasemap](#) 中该选项的介绍

**-M** 将边界数据以多段 ASCII 表的形式导出到标准输出

使用该选项，则只导出数据而不绘图，该选项需要与 **-E|-I|-N|-W** 选项一起使用。

**-N<border>[/<pen>]** 绘制政治边界。

该选项在某些地方与 **-E** 选项有重叠。边界类型 **<border>** 可以取：

- 1 : 国界
- 2 : 州界；(目前只有美国、加拿大、澳大利亚以及南美各国的数据)
- 3 : Marine boundaries
- a : 1-3 的全部边界；

说明：

1. **<border>** 是必须值，**<pen>** 是可选值
2. 可以多次重复使用 **-N** 选项，指定不同级别的边界
3. **<pen>** 的默认值是 **default,black,solid**

**-Q** 关闭区域裁剪。

使用 **-Gc** 和 **-Sc** 可以分别裁剪出 dry 区域和 wet 区域，接下来的其他绘图命令中只有在裁剪区域内的部分才会被绘制。在绘图结束后，需要关闭裁剪，就需要再次调用 **pscoast**，并加上 **-Q** 选项。若在开启裁剪后使用了 **-X** 和 **-Y** 选项，则在关闭时也要记得使用 **-X** 和 **-Y**。

**-S** 设置 wet 区域的填充色或裁剪 wet 区域

1. **-G<fill>** 设置 wet 区域（一般指海洋或湖泊）的填充色
2. **-Gc** 将 wet 区域裁剪出来，使得接下来的绘图只有 wet 区域内的才会被绘制

**-T** 绘制方向玫瑰图或磁场玫瑰图，见 [psbasemap](#) 中的选项介绍

**-W[<level>]/<pen>** 绘制湖岸线 (shoreline)。

shore 指水与陆地交界的“岸”（如：海岸、湖岸、河岸等），是一个较为笼统的说法。

GMT 将 shoreline 分成四个等级 (**<level>** 取 1-4)：

1. coastline：海岸线
2. lakeshore：湖泊与陆地的岸线
3. island-in-lake shore：首先要有陆地，陆地中有一个湖，湖里有一个岛。即岛的岸线
4. lake-in-island-in-lake shore：首先有陆地，陆地中有一个湖，湖中有一个岛，岛里又有一个湖。这里指的是湖的岸线

使用时需要注意：

1. 不使用 **-W** 选项，则不绘制任何 shore
2. 使用 **-W**，给定画笔属性 **<pen>**，但不给出 **<level>**，则绘制四个 level 的 shore

3. 可以用 `-W<level>/<pen>` 的方式指定要绘制哪一个 level 的 shore，并指定线条属性，在同一个命令中可以多次使用 `-W`，以指定不同 level 的 shore 的画笔属性
4. `-W` 选项中 `<level>` 是可选的，而 `<pen>` 是必须的！因而 `-W2` 会被解释为所有 level 的画笔属性，而不是 level 2

## 示例

```
gmt pscoast -R-30/30/-40/40 -Jm0.1i -B5 -I1/1p,blue -N1/0.25p,- \
            -I2/0.25p,blue -W0.25p,white -Ggreen -Sblue -P > africa.ps
```

```
gmt pscoast -R-30/-10/60/65 -Jm1c -B5 -Gp100/28 > iceland.ps
```

将非洲区域裁剪出来，并在其中的陆地部分绘制地形：

```
gmt pscoast -R-30/30/-40/40 -Jm0.1i -B5 -Gc -P -K > africa.ps
gmt grdimage -Jm0.1i etopo5.nc -Ccolors.cpt -O -K >> africa.ps
gmt pscoast -Q -O >> africa.ps
```

绘制部分国家的国界线（似乎有 BUG）：

```
gmt pscoast -JM6i -P -Baf -EGB,IT,FR+gblue+p0.25p,red+r -EES,PT,GR+gyellow > map.ps
```

提取冰岛的海岸线数据：

```
gmt pscoast -R-26/-12/62/68 -Dh -W -M > iceland.txt
```

## FAQ

1. 错误消息：

```
pscoast: low resolution shoreline data base not installed.
```

出现该错误的原因有如下几种：

- (a) 未安装 GSHHG 海岸线数据
- (b) 安装了但路径不正确（建议的做法是把所有 GSHHG 的文件放在 `$GMTHOME/share/coast` 目录下）
- (c) 安装的 netCDF 版本号为 3.x 而不是 4.x
- (d) 自行编译了 netCDF 4.x，且编译时使用了 `--disbale-netcdf4` 选项

## BUGS

1. -E 选项有尚未确定的 BUG (v5.2.1)

---

**pscontour**

---

官方文档 [pscontour](#)

---

## psconvert

---

官方文档 [psconvert](#)

**简介** 将 GMT 生成的 PS 文件转换为其他图片格式

支持 BMP、EPS、JPEG、PDF、PNG、PPM、SVG、TIFF 格式。原名为 `ps2raster`，v5.2.1 版本起更名为 `psconvert`

## 最小示例

如下命令会将 PS 文件转换成 JPG 格式:

```
gmt psconvert test.ps
gmt psconvert test1.ps test2.ps map*.ps
```

输出文件的文件名与原文件相同，仅后缀不同。

## 选项

**psfiles** 要转换格式的 PS 文件名

默认情况下，转换后的文件与转换前的文件有相同的文件名，文件后缀由文件格式决定

**-A** [**u**] [**<argins>**] [**-**] [**+g** **<paint>**] [**+p** **<pen>**] [**+r**] [**+s** [**m**] **|S** **<width>** [**<u>**]/**<height>** [**<u>**]] 对输出的图片做裁边

默认情况下，转换得到的图片的大小由 PS 文件的纸张尺寸决定。通常画图的时候是不会把一张 A4 纸画满的，所以在图片周围就会出现多余的白色部分。

**-A** 选项会对 PS 文件进行裁剪，仅保留其中有绘图的部分:

```
gmt psconvert -A test.ps
```

-Au 会先将 GMT 的时间戳去掉再裁剪。

默认的裁剪会把图片裁剪到尽可能小，感觉有些过了，可以加上额外的边距 <margins> 。其中 <margins> 可以取：

1. 一个数，四条边的额外边距相同，如 -A0.5c
  2. 两个数字，分别指定 X 和 Y 方向的额外边距，如 -A0.5c/1c
  3. 四个数字，分别指定上下左右四条边的边距，如 -A0.5c/0.5c/0.5c/0.5c
- -A+sm set a maximum size and the new width are only imposed if the original figure width exceeds it. Append <new\_height> to also impose a maximum height in addition to the width.
  - -A+s<width> 指定最终生成的图片的宽度，高度自动决定。程序会对图片做插值以保证 -E 的值
  - -A+S<scale> 则指定图片的缩放比例

如下命令会生成一个宽均为 5 厘米的图片：

```
gmt psconvert -A+s5cc test.ps
```

- -A- 覆盖 -W 选项中自动设置的 -A 值
- -A+r 会使得在计算边界时使用 round 函数而不是 ceil 函数，这会对裁剪造成极其微小的区别，仅当要处理非常小的图片时才需要使用。
- -A+g<paint> 为 BoundingBox 指定填充色
- -A+p<pen> 为 BoundingBox 指定边框颜色

-C<gs\_options> 要传递给 gs 的选项

该选项用于在调用 ghostscript 时传给 ghostscript 额外的选项，若要额外给 ghostscript 增加多个选项，可重复使用 -C 命令。

在 Windows 下，若 PS 文件中含中文，则可能需要使用 -C 选项告诉 ghostscript 字体路径：

```
gmt psconvert -C-sFONTPATH=C:\Windows\Fonts chinese.ps
```

-D<outdir> 设置输出目录

默认情况下，会在 PS 文件同一目录中生成其他图片文件，使用 -D<outdir> 选项可以指定输出目录，-D. 表示在当前目录输出。需注意，输出目录必须已存在，否则会报错。

-E<dpi> 设置图片精度

值越大，图片越清晰，文件也越大。PDF 格式默认值为 720，其他格式默认值为 300，单位为 dpi。

```
gmt psconvert -Tj -E600 test.ps
```

说明：

1. EPS 是矢量图片格式，-E 选项对其无效



2. PDF 是矢量图片格式, -E 选项对其中的 pattern 和字体有效

**-F<out\_name>** 指定输出的文件名

默认情况下, 该命令会直接使用输入的 PS 文件名, 并修改其后缀作为输出文件的文件名。比如 `test.ps` 转换出的图片则为 `test.jpg`、`test.png` 之类的。

-F 选项可强制指定输出文件名, 文件后缀由输出的文件格式自动决定:

```
gmt psconvert -Tf -Fout test.ps
```

**-G<gs\_path>** 指定 ghostscript 可执行文件的路径

psconvert 在底层是调用 ghostscript 来实现 PS 到其他格式的转换的, 因而成功转换的前提是必须能够找到 ghostscript 的可执行文件, -G 选项即用于显式指定 ghostscript 可执行文件的路径。

说明:

1. Linux 下一般不需要设置 ghostscript 的路径, 除非你自己重新编译了 gs 并安装到了非标准路径下
2. Windows 下, 一般也不需要使用该选项, 程序会自动从注册表里获取路径信息
3. 如果从注册表中获取路径失败, 则必须指定-G 选项, 例如  
`-Gc:\programs\gs\gs9.02\bin\gswin64c`

**-I** 见官方文档

**-L<listfile>** <listfile> 中包含要转换的所有 PS 文件名

**-P** 强制转换后的图片为 Portrait 模式。

若 PS 文件在绘图时使用了 -P 选项, 则 psconvert 的 -P 选项无效; 若 PS 文件在绘图时未使用 -P 选项, 则 psconvert 的 -P 选项会将图片旋转 90 度。

**-Q[gl t][1|2|4]** 见官方文档

**-S** 在执行 ghostscript 命令后, 将具体的命令打印到标准错误流中, 且保留转换过程中的所有临时文件。该选项主要用于调试。

**-Tb|e|E|f|F|j|g|G|m|s|t** 指定要转换的图片格式。可以接如下值:

- **b** : BMP;
- **e** : EPS;
- **E** : 带有 PageSize 命令的 EPS;
- **f** : PDF;
- **F** : 多页 PDF;
- **j** : JPEG (默认值);
- **g** : PNG;

- **G** : 透明 PNG;
- **m** : PPM;
- **s** : SVG;
- **t** : TIFF;

说明:

1. **g** 和 **G** 的区别在于前者背景色为白色, 后者背景色为透明;
2. 对于 **bjgt** , 可以在其后加 **-** , 则会将 PS 文件转换为灰度图;
3. EPS 格式可以与其他格式合在一起使用。比如 **-Tef** 会同时生成 EPS 和 PDF 文件。除此之外, 该命令一次只能转换一种格式, 比如 **-Tbf** 则只会生成 PDF 格式 (即以 **-T** 选项中的最后一个格式为准)
4. **-TF** 会将多个 PS/PDF 文件转换并合并成一个多页的 PDF 文件, 需要使用 **-F** 选项指定输出的文件名

转换为 PDF 格式:

```
gmt psconvert -Tf test.ps
```

转换为 JPG 格式的灰度图:

```
gmt psconvert -Tj- test.ps
```

利用一堆 PS 文件生成一个多页 PDF:

```
gmt psconvert -TF -Fout.pdf *.ps
```

**-W** 见官方文档

**-Z** 转换完成后删除输入的 PS 文件。若转换失败, 输入的 PS 文件不会被删除。

## 其他

1. 转换为 PDF、PNG 时使用 DEFLATE 压缩; 转换为 TIFF 时使用 LZW 压缩;
2. **psconvert** 还可以用于其他命令生成的 PS 文件;

## FAQ

1. 错误信息:

```
GMT PS format detected but file is not finalized. Maybe a -K in excess? No output created.
```

出现这种错误的最常见原因是最后一个绘图命令里多了一个 `-K`，即本该只使用 `-O` 选项关闭 PS 文件的，但却使用了 `-K -O` 选项。

## BUGS

1. Windows 下会自动为文件名加双引号，导致无法找到要转换的 PS 文件 (v5.1.1)
2. 使用 `-A` 选项时，`-C` 选项无效，导致无法在 Windows 转换含中文的 PS 文件 (v5.1.1)
3. 不能使用 `-F` 选项，即不能生成多页 PDF (v5.1.2)
4. `-Z` 选项无效 (v5.2.1)

---

## pscoupe

---

官方文档 [pscoupe](#)

简介 绘制震源机制解的剖面图

[psmeca](#) 在绘制震源球时，本质上是取了一个水平剖面，并将三维震源球的下半球投影到该水平剖面上。而 [pscoupe](#) 则更灵活一些，可以将三维震源球投影到任意一个剖面上。

- 对于一个水平剖面，会将下半球投影到平面上（即 [psmeca](#) 的做法）
- 对于一个垂直剖面，会将垂直平面后的半球投影到平面上
- 对于任意一个非水平的平面而言：
  - 北方向为平面的最速下降方向
  - 东方向为平面的走向方向
  - 下方向则根据右手定则确定

## 剖面类型

-A 选项用于确定剖面。

-Aa<lon1>/<lat1>/<lon2>/<lat2>/<dip>/<p\_width>/<dmin>/<dmax>[f]

- <lon1>/<lat1> 剖面起点的经纬度
- <lon2>/<lat2> 剖面终点的经纬度
- 起点和终点确定了剖面的长度
- <dip> 剖面所在平面的倾角（0 表示水平剖面，90 表示垂直剖面）
- <p\_width> 剖面的宽度（即剖面不是一个平面，而是一个有厚度的长方体）
- <dmin>/<dmax> 是沿着最速下降方向（“北”方向）的最小、最大距离
- f 表示根据剖面的参数自动计算边框的范围

- TODO: `<p_width>/<dmin>/<dmax>` 的具体含义尚不明确

`-Ab<lon1>/<lat1>/<strike>/<p_length>/<dip>/<p_width>/<dmin>/<dmax>[f]`

- `<lon1>/<lat1>` 剖面起点的经纬度
- `<strike>` 是剖面的走向
- `<p_length>` 是剖面的长度
- 其他参数与 `-Aa` 相同

`-Ac<x1>/<y1>/<x2>/<y2>/<dip>/<p_width>/<dmin>/<dmax>[f]`

与 `-Aa` 选项相同，只是 `<x>/<y>` 为笛卡尔坐标而不是地理坐标

`-Ad<x1>/<y1>/<strike>/<p_length>/<dip>/<p_width>/<dmin>/<dmax>[f]`

与 `-Ab` 选项相同，只是 `<x>/<y>` 为笛卡尔坐标而不是地理坐标

## 选项

`-E<color>` 扩张部分的填充色，默认为白色

`-F<mode>[<args>]` 设置多个属性，可重复使用多次

- `-Fs<symbol>[<size>/<fontsize>[<offset>[<u>]]]]` 见官方文档
- `-Fa[<size>][</P_axis_symbol>[<T_axis_symbol>]]` 计算并绘制 P 和 T 轴的符号。`<size>` 是符号大小，符号与 *psxy* 类似，可以取 `cl|d|h|i|p|s|t|x`，默认值为 `6p/cc`
- `-Fe<fill>` 设置 T 轴符号的填充色
- `-Fg<fill>` 设置 P 轴符号的填充色
- `-Fp<pen>` P 轴轮廓的画笔属性
- `-Ft<pen>` T 轴轮廓的画笔属性
- `-Fr<fill>` 在标签后加一个方框

`-G<color>` 指定压缩部分的填充色，默认值为黑色

`-L[<pen>]` 设置震源球外部轮廓的线条属性

`-M` 所有震级使用相同的大小，具体大小由 `-S` 选项的 `<scale>` 参数决定。

`-N` 地图区域外的震源球也要绘制，默认不绘制。

`-Q` 默认会生成一些临时文件，其中包含了剖面 and 剖面上的震源机制的信息，使用该选项，则不会生成这些临时文件。

`-T<num_of_planes>[</pen>]` 绘制断层平面。`<num_of_planes>` 可以取：

- 0 绘制两个断层面

- 1 绘制第一个断层面
- 2 绘制第二个断层面

**-S** 选择震源机制解的格式。与 *psmeca* 中 **-S** 选项用法相同

唯一的不同在于，输入文件中 **newX** 和 **newY** 对该命令无效，这里加上 **newX** 和 **newY** 只是为了是的该命令可以使用与 *psmeca* 相同的输入文件。

**-W<pen>** 设置断层边界的画笔属性

**-Z<cpt>** 指定 CPT 文件，根据数据文件中第三列的值（即地震深度）确定震源球的压缩部分的颜色。

## 示例

下面的示例展示了 **-Aa** 选项的用法，指定 P1 和 P2 为剖面的起点和终点，并展示了不同倾角时的震源球的效果。

```
#!/bin/bash
ps=pscoupe_ex1.ps

gmt gmtset PROJ_LENGTH_UNIT inch MAP_TICK_LENGTH_PRIMARY 0.075i \
    MAP_FRAME_WIDTH 0.1i MAP_ORIGIN_X 2.5c MAP_ORIGIN_Y 1.3i

# Plotting 2 mechanisms on map
gmt psmeca -R128/130/10/11.1 -JX2i -Fa0.1i/cc -Sc0.4i -B1 -Y8.5i -P -K << EOF > $ps
# lon   lat  dep str dip rake str dip rake m ex nx ny
129.5   10.5 10   0   90   0  90   90 180   1 24   0   0 10km
128.5   10.5 40   0   45   90 180   45  90   1 24   0   0 40km
EOF
(echo 128 11; echo 130 11) | gmt psxy -R -J -K -O -W0.25p,red >> $ps
gmt pstext -R -J -N -F+f14p,Helvetica-Bold+j -K -O << EOF >> $ps
128 11 ML P1
130 11 MR P2
EOF

# Represent cross-sections between points P1(128E11N) and P2(130E/11N)
# on a plane the dip of which varies
# from quasi horizontal to vertical.
# y dimension is counted along steepest descent on the plane
# so the values of depth are only in the vertical cross-section.

# Variation of dip for cross-section plane
# (WE azimuth, between points (128E,11N) and (130E,11N))

plots () {
```

```

y_offset=-2.5i
for d in $1 $2 $3 ; do
    gmt pscoupe -R0/200/0/100 -JX1.5i/-1.5i -Bxa100f10 -Bya50f10 -BWesN \
        -Q -L -Sc0.4 -Aa128/11/130/11/$d/60/0/100f -Ggrey -Fa0.1i/cc $4 $5 \
        -Y$y_offset -X$x_offset -O -K << EOF
# lon  lat  dep str dip rake str dip rake m ex nx ny
129.5 10.5  10  0   90   0  90   90 180  1 24  0  0 10km
128.5 10.5  40  0   45  90 180   45  90  1 24  0  0 40km
EOF
    gmt pstext -R -J -F+f18p,Helvetica-Bold+jBL -O -K <<< "10 15 $d"
    y_offset=0i
    x_offset=2.5i
done
x_offset=-5i
}

x_offset=0i

plots 10 20 30 >> $ps
plots 40 50 60 >> $ps
plots 70 80 90 -N >> $ps

gmt pstext -X-5i -R0/10/0/15 -Jx1i -F+jBL+fHelvetica-Bold+f -O << EOF >> $ps
3 8.5 24 Variation of dip
3 8.0 20 W-E cross-section
EOF
rm gmt.*

```

下面的示例将震源球投影到不同方位角的剖面上的效果：

```

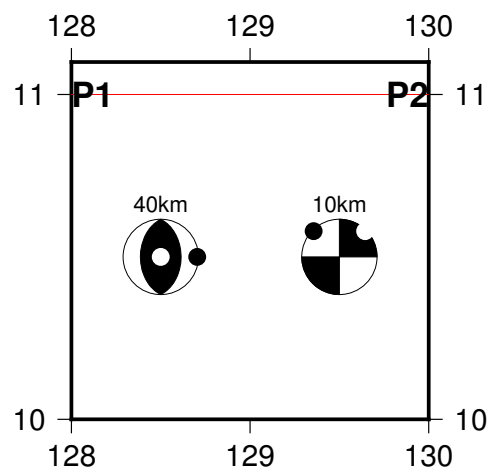
#!/bin/bash
# test/meca/meca_4.sh

ps=pscoupe_ex2.ps

gmt gmtset PROJ_LENGTH_UNIT inch MAP_TICK_LENGTH_PRIMARY 0.075i MAP_FRAME_WIDTH 0.1i \
    MAP_ORIGIN_X 2.5c MAP_ORIGIN_Y 1.3i

# Plotting 2 mechanisms on map
gmt psmeca -R128/130/10/11.1 -JX2i -Fa0.1i/cc -Sc0.4i -B1 -Y8.5i -P -K << EOF > $ps
# lon  lat  dep str dip rake str dip rake m ex nx ny
129.5 10.5 10  0   90   0  90   90 180  1 24  0  0 10km
128.5 10.5 40  0   45  90 180   45  90  1 24  0  0 40km
EOF
(echo 128 11; echo 130 11) | gmt psxy -R -J -K -O -W0.25p,red >> $ps

```



## Variation of dip

### W-E cross-section

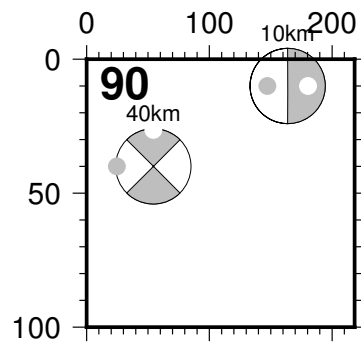
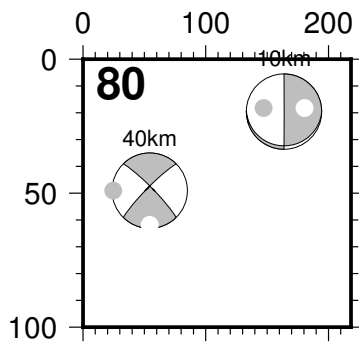
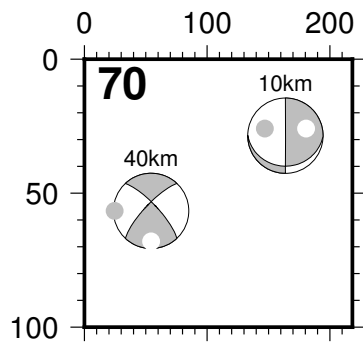
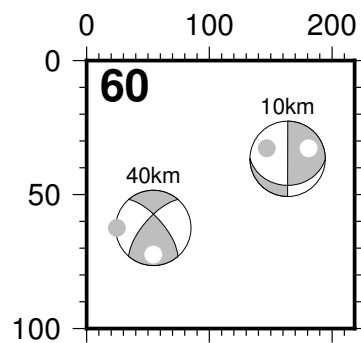
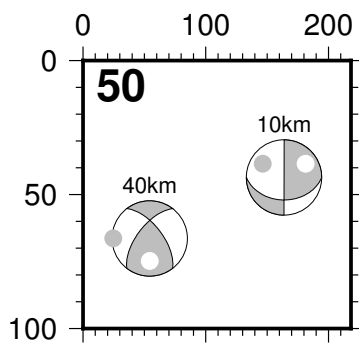
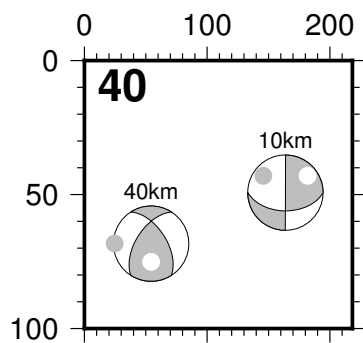
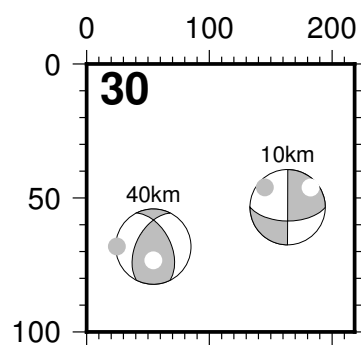
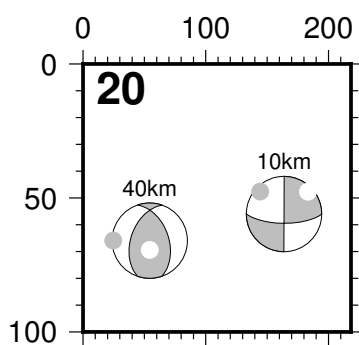
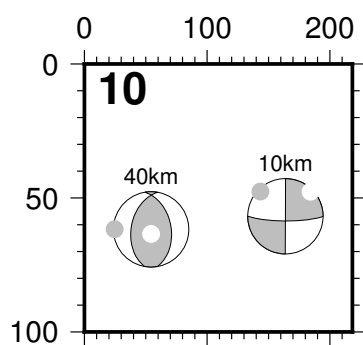


图 88.1: pscoupe 示例图 1



```

gmt pstext -R -J -N -F+f14p,Helvetica-Bold+j -K -O << EOF >> $ps
128 11 ML P1
130 11 MR P2
EOF

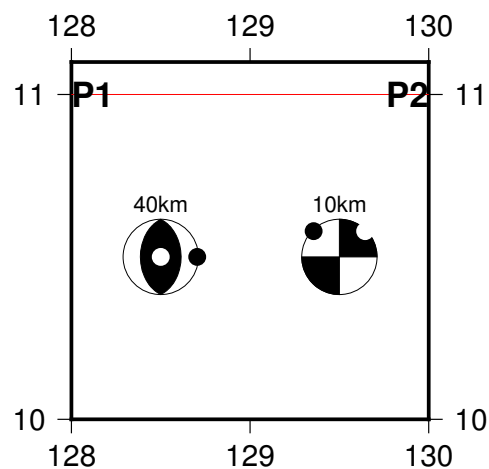
plots () {
y_offset=-2.5i
for a in $1 $2 $3 ; do
    gmt pscoupe -R0/250/0/100 -JX1.5i/-1.5i -Bxa100f10 -Bya50f10 -BWesN \
        -Q -L -Sc0.4 -Ab$4/$5/$a/250/90/$6/0/100f -Ggrey -Fa0.1i/cc $7 $8 \
        -Y$y_offset -X$x_offset -O -K << EOF
# lon lat dep str dip rake str dip rake m ex nx ny
129.5 10.5 10 0 90 0 90 90 180 1 24 0 0 10km
128.5 10.5 40 0 45 90 180 45 90 1 24 0 0 40km
EOF
    gmt pstext -R -J -F+f18p,Helvetica-Bold+jBR -O -K <<< "240 90 $a"
    y_offset=0i
    x_offset=2.5i
done
x_offset=-5i
}

x_offset=0i

plots 0 40 80 128 10.0 200 >> $ps
plots 120 160 200 128 11.0 400 -N >> $ps
plots 240 280 320 130 10.5 200 -N >> $ps

gmt pstext -X-5i -R0/10/0/15 -Jx1i -F+jBL+fHelvetica-Bold+f -O << EOF >> $ps
3 8.5 24 Variation of azimuth
3 8.0 20 vertical cross-section
EOF
rm gmt.*

```



## Variation of azimuth vertical cross-section

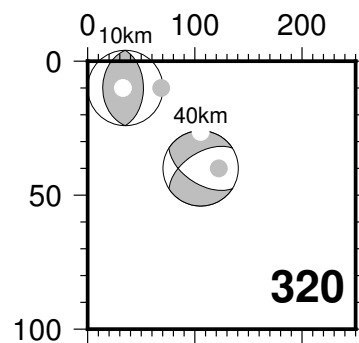
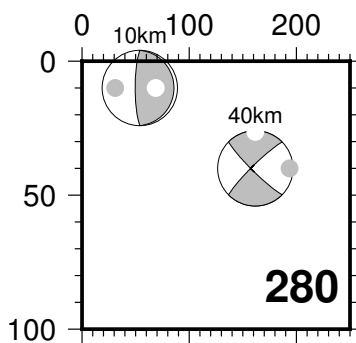
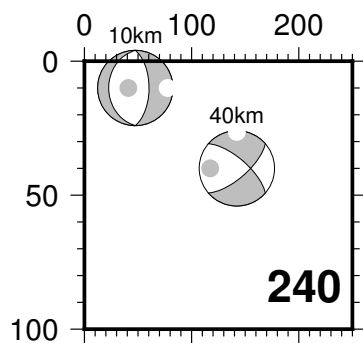
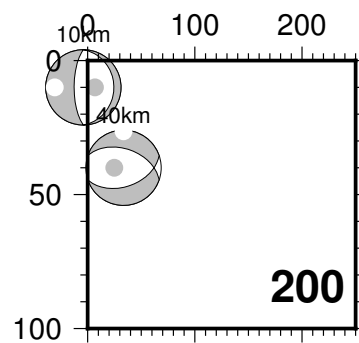
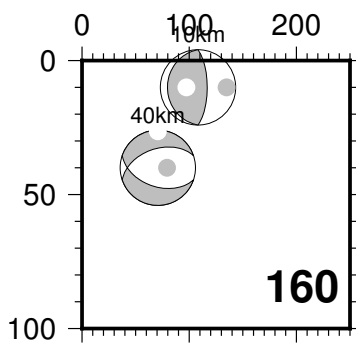
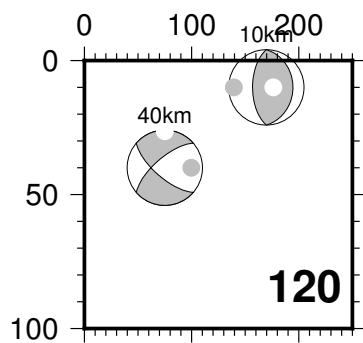
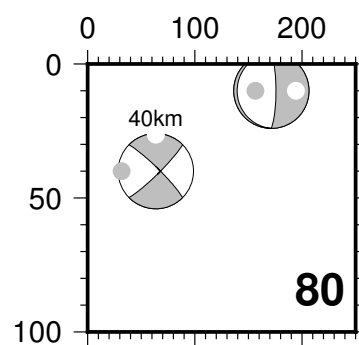
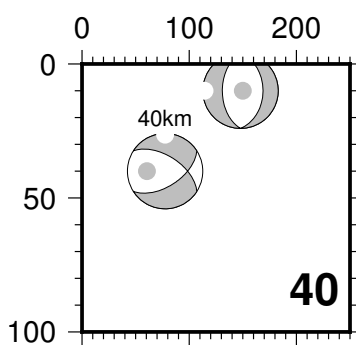
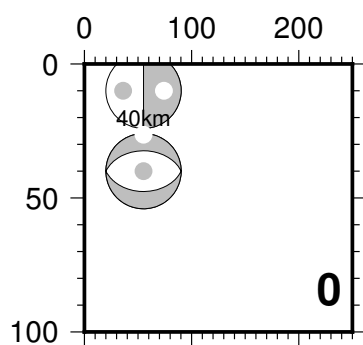


图 88.2: pscoupe 示例图 2

---

## pshistogram

---

官方文档 [pshistogram](#)

简介 统计并绘制直方图

该命令会读取数据中的第一列，对其进行统计，并绘制直方图或累积直方图。

## 选项

**-W<bin\_width>[+l|h|b]** 设置直方图计算时的 bin 宽度

[+l|h|b] 用于控制落在 -R 范围外的数据点的处理方式：

- 默认情况下，这些数据会被忽略
- +b 将落在范围外的数据包含在第一个或最后一个 bin 中
- +l 仅将小于第一个 bin 的数据放在第一个 bin 中
- +h 仅将大于最后一个 bin 的数据放在最后一个 bin 中

**-A** 从 X=0 处开始绘制水平直方图，默认是从 Y=0 处绘制垂直直方图

**-C<cpt>** 指定 CPT 文件，将每个 bar 的中间值作为 Z 值查询 CPT 中的颜色

**-D[+b][+f<font>][+o<offset>][+r]** 为每个 bar 添加标注，标注内容是每个 bar 的统计数目

- +b 将标注放在 bar 的下边（默认放在上边）
- +f<font> 设置标注的字体
- +o<offset> 修改 bar 与标注的距离
- +r 将标注从水平方向旋转为垂直方向

**-F** center bin on each value（默认是左边界）

假设数据范围是 0 到 100, bin 宽度为 10。默认情况下, 会将 0 到 10 作为第一个 bin, 10 到 20 作为第二个 bin, 以此类推。若使用该选项, 则第一个 bin 以 0 为中心, 即 0 到 5 是第一个 bin, 5 到 15 是第二个 bin, 以此类推。

**-G<fill>** 设置 bar 的填充色

**-I[o|O]** 返回计算结果不绘图。

- **-I** 返回 `xmin xmax ymin ymax`
- **-Io** 返回 bin 的 X 和 Y 值
- **-IO** 返回 bin 的 X 和 Y 值, 即便 `Y=0`

**-L<pen>** 设置 bar 的轮廓属性

**-N[<mode>][+p<pen>]** 绘制等效的正态分布曲线

**<mode>** 用于设定正态分布的中间位置及比例:

- `mode=0`: mean and standard deviation
- `mode=1`: median and L1 scale
- `mode=2`: LMS mode and scale

**<pen>** 用于指定曲线的属性。该选项可以使用多次以绘制多条曲线。

**-Q** 绘制累积直方图

**-S** 绘制阶梯状直方图。

默认的直方图中, 每个 bin 都用一个 bar 表示。使用该选项, 则去除 bar 内部的线条。

**-Z<type>** 选项直方图的种类

- `type=0`: counts (默认值)
- `type=1`: frequency\_percent
- `type=2`:  $\log(1.0 + \text{count})$
- `type=3`:  $\log(1.0 + \text{frequency\_percent})$
- `type=4`:  $\log_{10}(1.0 + \text{count})$
- `type=5`:  $\log_{10}(1.0 + \text{frequency\_percent})$ .

若需要绘制其他种类的直方图, 则只能先自己对输入数据做处理再绘图。

## psimage

官方文档 [psimage](#)

**简介** 将图片或 EPS 文件放在地图上

该命令可以读取一个 EPS 文件或其他图片格式（不包含 PS 格式的文件），并将其画在地图上，可以用于将自己单位的 Logo 放在 GMT 做的图上。

## 选项

**<imagefile>** EPS 文件或其他光栅图片格式（GIF、PNG 等）的文件

- EPS 文件必须包含合适的 BoundingBox
- 光栅文件的颜色深度可以是 1、8、24、32 位
- 光栅文件是通过 GDAL 读入的，若安装 GMT 时未配置 GDAL，则该命令只能读取 Sun 光栅文件

**-D[g|j|J|n|x]<refpoint>+r<dpi>+w[-]<width>[/<height>][+j<justify>][+n<nx>[/<ny>]][+o<dx>[/<dy>]]**  
指定图片的尺寸和位置。

1. [g|j|J|n|x]<refpoint> 指定底图上的参考量，见 [embellishments](#) 一节
2. +j<justify> 指定色标上的锚点，默认锚点是 BL，见 [embellishments](#) 一节
3. +o<dx>[/<dy>] 指定参考点的额外偏移量，见 [embellishments](#)
4. +r<dpi> 指定图片的 DPI 以间接指定图片的尺寸
5. +w[-]<width>[/<height>] 直接指定图片的尺寸，若未给定 <height> 则按照 <width> 以及原图的纵横比进行缩放；若 <width> 为负值，则使用其绝对值作为宽度，并使用 PS 的图片操作符将图片插值到设备的分辨率
6. +n<nx>[/<ny>] 使图片在水平方向重复 <nx> 次，垂直方向重复 <ny> 次，若省略 <ny> 则默认其与 <nx> 相等，默认值为 1/1

**-F** 为图片加上背景边框，见 [embellishments](#) 一节

**-M** 使用 YIQ 变换将彩图转换成灰度图

**-G[b|f|t]<color>** 对图片颜色的一些设置

1-bit 图片默认为黑色和白色，可以通过如下选项进行修改：

- **-Gb<color>** 设置背景色，即将白色替换成其他颜色
- **-Gf<color>** 设置前景色，即将黑色替换成其他颜色
- **<color>** 可以取 **-**，表示透明

对于 8、24、32 位图片而言：

- **-Gt<color>** 将某个特定颜色设置为透明

**-I** 绘图前对 1-bit 图片进行反转，即黑色变白色，白色变黑色

## 示例

```
gmt psimage logo.jpg -Dx0/0+w1i -F+pthin,blue > image.ps
```

```
gmt psimage tiger.eps -Dx2i/1i+jTR+w3i > image.ps
```

```
gmt psimage 1_bit.ras -Gbbrown -Gfred -Dx0/0+w1c+n5 > image.ps
```

## pslegend

官方文档 [pslegend](#)

简介 在图上添加图例

## 选项

**-C<dx>/<dy>** 设置图例边框与内部图例之间的空白，默认值为 4p/4p

**-D[g|j|J|n|x]<refpoint>+w<length>/<width>[+j<justify>][+l<spacing>][+o<dx>/<dy>]]** 设置图例的位置和大小

- [g|j|J|n|x]<refpoint> 指定底图上的参考点，见 [embellishments](#)
- +j<justify> 指定图例上的锚点，默认锚点是 BL，见 [embellishments](#) 一节
- +o<dx>/<dy> 指定参考点的额外偏移量
- +w<length>/<width> 用于指定图例框的尺寸，若 <height> 未指定或取 0，则 GMT 会自动估算所需的高度
- +l<spacing> 行间距因子，默认值为 1.1，则当前字体大小的 1.1 倍

该选项几个比较有用的用法是：

1. 将图例放在左下角：-DjBL+w4c/2c+o0.2c/0.2c
2. 将图例放在左上角：-DjTL+w4c/2c+o0.2c/0.2c
3. 将图例放在右下角：-DjBR+w4c/2c+o0.2c/0.2c
4. 将图例放在右上角：-DjTR+w4c/2c+o0.2c/0.2c

**-F** 控制图例的背景属性，见 [embellishments](#) 一节

默认图例无边框，使用该选项则会给图例绘制边框。

## 图例文件格式

图例文件用于控制图例中各项的布局。图例文件中的每个记录对应图例中的一项，图例中每项的顺序由记录的先后顺序决定。每个记录的第一个字符决定了当前记录的图例类型。GMT 中共有 14 种图例类型，列举如下：

**# comment** 以 # 开头的行或空行都会被跳过

**A cptname** 指定 CPT 文件，使得某些记录可以通过指定 Z 值来设定颜色，可以多次使用该记录以指定不同的 CPT 文件

**B cptname offset height [optional arguments]** 绘制水平 colorbar

1. **offset** 是 colorbar 相对于图例框左边界的距离
2. **height** 是 color 高度
3. 还可以加上子选项 **+e[b|f][<length>][+h][+ma|c|l|u][+n[<txt>]** 以及其他一些参数：-B、-I、-L、-M、-N、-S、-Z、-p 等，详情见 [psscale](#) 命令

**C textcolor** 接下来的所有文本所使用的颜色。

可以直接指定颜色，也可以用 **z=<val>** 指定 Z 值，以从 CPT 文件中查找相应的颜色（CPT 文件由 A 记录指定），若 **textcolor** 为 -，则使用默认颜色

**D [offset] pen [-l+l=]** 绘制一条水平线

1. **offset** 为线条左右顶端与图例边框的空白距离
2. **pen** 为线条属性
  - (a) 默认值为 0
  - (b) 若未指定 **pen**，则使用 `MAP_GRID_PEN_PRIMARY`
  - (c) 若 **pen** 设置为 -，则绘制一条不可见的线
3. 默认情况下，线条上下各留出四分之一的行间距，**-l+l=** 分别表示线条上方无空白、线条下方无空白和线条上下均无空白。

**F fill1 fill2 ... filln** 指定单元的填充色。

可以直接指定颜色，也可使用 **z=<value>** 形式指定从 CPT 文件中查找颜色。若只给定了一个 **fill**，则整行都使用相同的填充色，否则依次为当前行的每列应用不同的 **fill**，若 **fill** 为 -，则不填充。

**G gap** 给定一个垂直空白

空白的高度由 **gap** 决定，**gap** 可以用 **ilc|p** 单位，也可以用 **l** 作为单位表示多少行空白，**gap** 也可以取负值，表示将当前行上移。

**H fontsize|-font|-header** 为图例指定一个居中的标题。

1. **header** 为标题
2. **fontsize** 为字号



3. `font` 为字体号

4. `-` 表示使用默认的文字大小以及默认字体 `FONT_TITLE`

**I** `imagefile width justification` 将 EPS 或光栅文件放在图例中

1. `width` 为图片宽度

2. `justification` 为图片的对齐方式

**L** `fontsize|-font|-justification label` 在图例中绘制标签。

1. `label` 为要绘制的标签

2. `fontsize` 为字号

3. `font` 为字体号

4. `justification` 为对齐方式，可以取 `L|C|R`

5. `-` 表示使用默认的文字大小以及默认字体 `FONT_TITLE`

**M** `slon|-slat length [+f][+l[label]][+u] [-Fparam] [-Rw/e/s/n -Jparam]` 在图例中绘制比例尺，在 [psbasemap](#) 命令中有详细介绍

**N** `ncolumns` 或 **N** `relwidth1 relwidth2 ... relwidthn` 修改图例中的列数

默认只有一列，该记录仅对 **S** 和 **L** 记录有效。该记录指定的列数会一直有效直到再次使用 **N** 记录。  
`ncolumns` 用于指定若干个等宽的列，`relwidth1 relwidth2 ... relwidthn` 用于指定每列所占的相对宽度，所有宽度的和应等于 `-D` 选项所设置的宽度相等。

**P** `paragraph-mode-header-for-pstext` 在图例中添加段落，参考 [pstext](#) 命令中的段落模式

**S** `[dx1 symbol size fill pen] [dx2 text]` 在图例中绘制符号

1. `symbol` 指定了符号类型，见 [psxy](#) 命令

2. `dx1` 是符号中心与左边界的距离

3. `dx2` 是 `text` 与左边界的距离

4. `text` 的字体由参数 `FONT_ANNOT_PRIMARY` 控制

5. `fill` 和 `pen` 控制符号的填充和轮廓颜色，设置为 `-` 则表示不填充或无轮廓颜色

6. `fill` 可以使用 `z=<val>` 的形式，从 CPT 文件中查找颜色

7. `dx1` 除了可以指定距离，还可以使用 `L|C|R` 表示符号在当前列的对齐方式

8. 若 **S** 记录中无其他参数，则直接调至下一列

9. 若 `symbol` 取为 `f|q|v`，则可以在 `symbol` 后加上额外的子选项

10. 符号 `-` 可以用于绘制线段

11. 某些符号可能需要指定多个 `size`，将多个 `size` 用逗号分隔作为 `size` 即可

**T** `paragraph-text` 用参数 `FONT_ANNOT_PRIMARY` 打印一段文本

V [offset] pen 在两列之间绘制垂直的线条

## 示例

```
#!/bin/bash
gmt pslegend -R-10/10/-10/10 -JM6i -F+gazure1 -Dx0.5i/0.5i+w5i/3.3i+jBL+l1.2 \
    -CO.1i/0.1i -B5f1 << EOF > pslegend_ex1.ps
# Legend test for pslegend
# G is vertical gap, V is vertical line, N sets # of columns, D draws horizontal line.
# H is header, L is label, S is symbol, T is paragraph text, M is map scale.
#
G -0.1i
H 24 Times-Roman My Map Legend
D 0.2i 1p
N 2
V 0 1p
S 0.1i c 0.15i p300/12 0.25p 0.3i This circle is hachured
S 0.1i e 0.15i yellow 0.25p 0.3i This ellipse is yellow
S 0.1i w 0.15i green 0.25p 0.3i This wedge is green
S 0.1i f0.1i+l+t 0.25i blue 0.25p 0.3i This is a fault
S 0.1i - 0.15i - 0.25p,- 0.3i A dashed contour
S 0.1i v0.1i+a40+e 0.25i magenta 0.25p 0.3i This is a vector
S 0.1i i 0.15i cyan 0.25p 0.3i This triangle is boring
V 0 1p
D 0.2i 1p
N 1
M 5 5 600+u f
G 0.05i
#I SOEST_logo.ras 3i CT
G 0.05i
B rainbow.cpt 0.2i 0.2i
G 0.05i L 9 4 R Smith et al., @5%J. Geophys. Res., 99@%, 2000
G 0.1i
P
T Let us just try some simple text that can go on a few lines.
T There is no easy way to predetermine how many lines will be required,
T so we may have to adjust the box height to get the right size box.
EOF
rm gmt.*
```

```
#!/bin/bash
ps=pslegend_ex2.ps
```

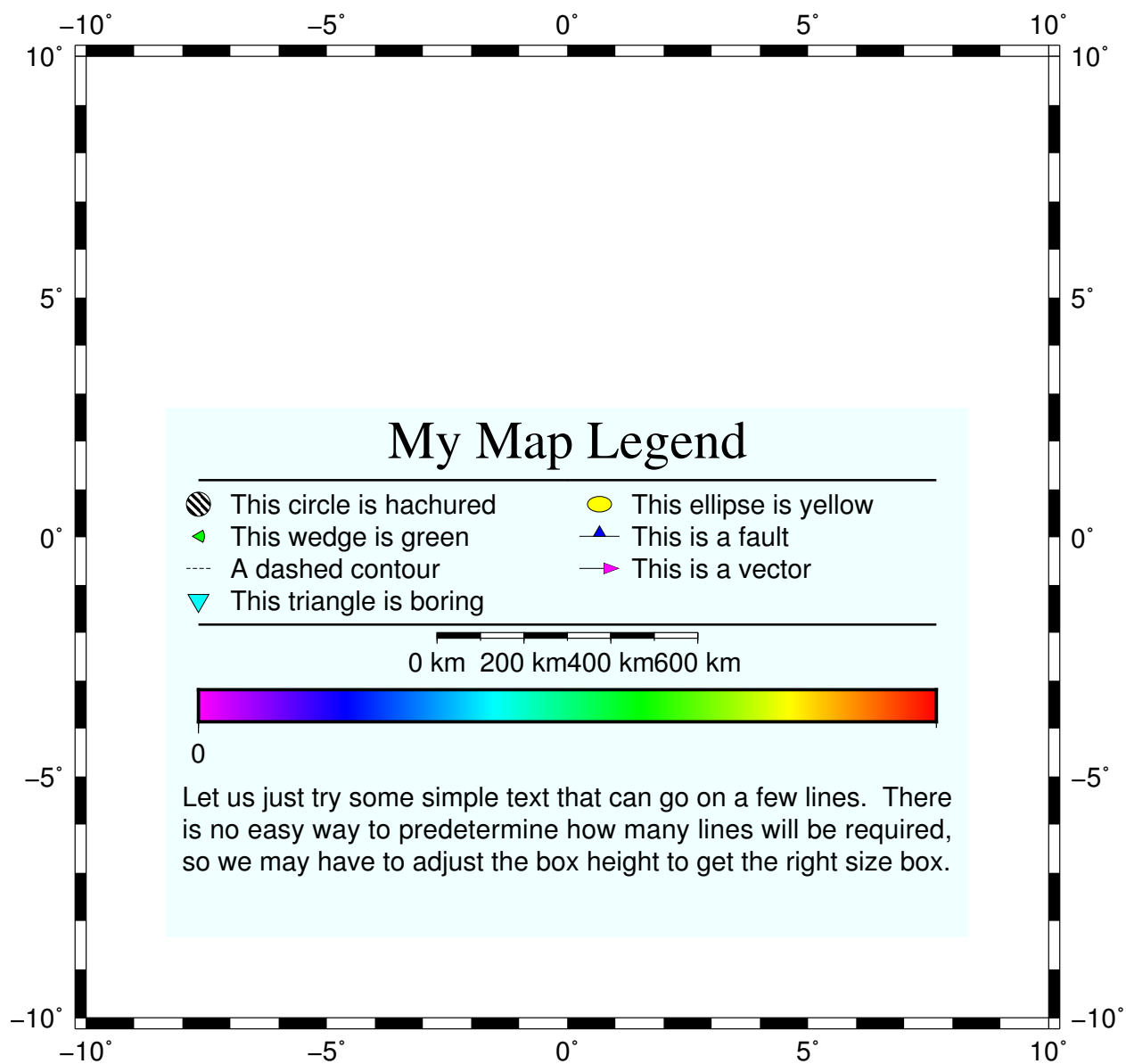


图 91.1: pslegend 示例图 1

```
gmt gmtset FONT_ANNOT_PRIMARY 12p FONT_LABEL 12p
```

```
cat <<EOF > table.txt
```

```
#G 0.04i
```

```
H 24 Times-Roman Eight Largest Cities in North America
```

```
D 1p
```

```
N 6 22 16 20 20 8 8
```

```
V 0.25p
```

```
S 0.15i c 0.1i snow1 -
```

```
L - - C City Name
```

```
L - - C Country
```

```
L - - C Population
```

```
L - - C Climate
```

```
L - - C WC?
```

```
L - - C OL?
```

```
D 0 1p
```

```
F lightgreen
```

```
S 0.15i c 0.1i red 0.25p
```

```
L - - R Mexico City
```

```
L - - R Mexico
```

```
L - - R 8,851,080
```

```
L - - R Tropical
```

```
L - - C Y
```

```
L - - C Y
```

```
F -
```

```
S 0.15i c 0.1i orange 0.25p
```

```
L - - R New York City
```

```
L - - R USA
```

```
L - - R 8,405,837
```

```
L - - R Tempered
```

```
L - - C Y
```

```
L - - C N
```

```
S 0.15i c 0.1i yellow 0.25p
```

```
L - - R Los Angeles
```

```
L - - R USA
```

```
L - - R 3,904,657
```

```
L - - R Subtropical
```

```
L - - C Y
```

```
L - - C Y
```

```
F lightblue
```

```
S 0.15i c 0.1i green 0.25p
```

```
L - - R Toronto
```

```
L - - R Canada
```

```
L - - R 2,795,060
```

```
L - - R Tempered
```

```

L - - C N
L - - C N
F -
S 0.15i c 0.1i blue 0.25p
L - - R Chicago
L - - R USA
L - - R 2,714,856
L - - R Tempered
L - - C Y
L - - C N
S 0.15i c 0.1i cyan 0.25p
L - - R Houston
L - - R USA
L - - R 2,714,856
L - - R subtropical
L - - C N
L - - C N
F lightred
S 0.15i c 0.1i magenta 0.25p
L - - R Havana
L - - R Cuba
L - - R 2,106,146
L - - R Tropical
L - - C N
L - - C N
F lightblue
S 0.15i c 0.1i white 0.25p
L - - R Montreal
L - - R Canada
L - - R 1,649,519
L - - R Tempered
L - - C N
L - - C Y
D 1p
V 1p
F -
N 1
L 9 4 R Information from Wikipedia
G 0.05i
T Many of these cities have hosted World Cup Soccer (WC) and some
T have hosted the Olympics (OL). The rest is just some basic information
T about each city, such as climate and population. Of course, this is all
T an excuse to demonstrate variable-width tables and row coloring.
#G 0.1i
EOF

```

```
cat << EOF > t.cpt
1      red
2      orange
3      yellow
4      green
5      blue
6      cyan
7      magenta
8      white
EOF
gmt pslegend -Dx0/0+w5.6i+jBL+l1.1 -C0.05i -F+p+gsnow1 -B0 table.txt -P --FONT_ANNOT_PRIMARY=12p \
--FONT_LABEL=12p > $ps
rm -f table.txt t.cpt gmt.*
```

Eight Largest Cities in North America						
	City Name	Country	Population	Climate	WC?	OL?
●	Mexico City	Mexico	8,851,080	Tropical	Y	Y
●	New York City	USA	8,405,837	Tempered	Y	N
●	Los Angeles	USA	3,904,657	Subtropical	Y	Y
●	Toronto	Canada	2,795,060	Tempered	N	N
●	Chicago	USA	2,714,856	Tempered	Y	N
●	Houston	USA	2,714,856	subtropical	N	N
●	Havana	Cuba	2,106,146	Tropical	N	N
●	Montreal	Canada	1,649,519	Tempered	N	Y

Information from Wikipedia

Many of these cities have hosted World Cup Soccer (WC) and some have hosted the Olympics (OL). The rest is just some basic information about each city, such as climate and population. Of course, this is all an excuse to demonstrate variable-width tables and row coloring.

图 91.2: pslegend 示例图 2

---

**psmask**

---

官方文档 [psmask](#)

官方文档 [psmeca](#)

简介 在地图上绘制震源机制解

## -S 选项

-S 确定了输入文件的格式，其语法为：

```
-S<format><scale>[/<fontsize>[/<offset>]] [u]]
```

- <format> 可以去 a 、 c 等，在下面会介绍
- <scale> 指定了 5 级地震（地震矩为 4.0E23 dynes-cm）的震源球的半径。默认情况下，震源球的半径与震级大小成正比，实际半径由如下公式决定：

```
size = M / 5 * <scale>
```

- <fontsize> 是震源球的标题的字体大小
- <offset> 震源球标题相对于震源球的偏移量
- u 震源球标题位于震源球的下方

下面介绍各种震源机制解的格式，使用下面这个地震作为示例：

```
Date: 1976/ 1/ 1   Centroid Time:  1:29:53.4 GMT
Lat= -29.25 Lon=-176.96
Depth= 47.8   Half duration= 9.4
Centroid time minus hypocenter time: 13.8
Moment Tensor: Expo=26  7.680 0.090 -7.770 1.390 4.520 -3.260
Mw = 7.3   mb = 6.2   Ms = 0.0   Scalar Moment = 9.56e+26
Fault plane: strike=202   dip=30   slip=93
Fault plane: strike=18   dip=60   slip=88
```



-Sa Aki and Richards 约定的震源机制格式，输入文件的具体格式为:

```
X Y depth strike dip rake mag newX newY title
```

- X 和 Y 为震源经纬度
- depth 是地震深度，单位为 km
- strike 、 dip 、 rake 断层的三个参数，单位为度
- mag 为震级
- newX 和 newY 放置震源球的经纬度，这样可以避免震源球覆盖震源位置。若值为 0，则表示将震源球放在震源位置
- title 出现中震源球上方/下方的字符串（可选）

```
gmt psmeca -Rg -JN15c -Sa2c > meca.ps << EOF
-176.96 -29.25 47.8 202 30 93 7.3 0 0 197601010129
EOF
```

-Sc Harvard CMT（现在的 Global CMT）定义的格式:

```
X Y depth strike1 dip1 rake1 strike2 dip2 rake2 mantissa exponent newX newY title
```

- X 和 Y 为震源经纬度
- depth 是地震深度，单位为 km
- 两组 strike 、 dip 、 rake 分别是两个断层面的参数
- mantissa 和 exponent 是地震标量矩的尾数和指数部分。例如，地震标量矩为  $9.56 \times 10^{26}$  dyne-cm，则 mantissa=9.56 、 exponent=26

```
gmt psmeca -Rg -JN15c -Sc2c > meca.ps << EOF
-176.96 -29.25 47.8 202 30 93 18 60 88 9.56 26 0 0 197601010129
EOF
```

-Sm|d|z GCMT 的零迹矩张量解，输入数据格式为:

```
X Y depth mrr mtt mff mrt mrf mtf exp newX newY title
```

- X 和 Y 为震源经纬度
- depth 是地震深度，单位为 km
- mrr 等是地震矩的 6 个分量，单位是  $10^{exp}$  dyne-cm
- exp 地震矩的指数部分（比如，mrr=2.0 、 exp=26 ，则真实的 mrr= $2.0 \times 10^{26}$  ）

地震矩张量可以分解成各向同性部分（ISO）、双力偶部分（DC）和补偿线性向量偶极部分（CLVD）。

- m 表示绘制地震矩的零迹部分 (DC+CLVD)
- d 表示仅绘制地震矩的双力偶部分 (DC)
- z 表示仅绘制地震矩的零迹各向异性部分 (DC+CLVD)

说明:

- m 和 z 是一个东西? 还是我的理解有误? TODO
- GMCT 所使用的 rtf 坐标, 实际上就是 USE 坐标系

```
gmt psmeca -Rg -JN15c -Sm2c > meca.ps << EOF
-176.96 -29.25 47.8 7.680 0.090 -7.770 1.390 4.520 -3.260 26 0 0 title
EOF
```

-Sp 由两个断层平面的部分数据构成的机制解, 输入数据格式为:

```
X Y depth strike1 dip1 strike2 fault mag newX newY title
```

- X 和 Y 为震源经纬度
- depth 是地震深度, 单位为 km
- strike1 和 dip1 平面 1 的断层参数
- strike2 平面 2 的断层参数
- fault 取-1 或 +1, 表示正断层和逆断层
- mag 震级

-Sx|y|t 指定主坐标轴的方位, 即 T、N、P 轴, 输入数据格式为:

```
X Y depth Tvalue Tazim Tplunge Nvalue Nazim Nplunge Pvalue Pazim Pplunge exp newX newY title
```

- X 和 Y 为震源经纬度
- depth 是地震深度, 单位为 km
- Tvalue 等 9 个量定义了 T、N、P 轴的大小和方向
- exp 是 Tvalue 等的指数部分

对这种表示不太熟悉, 应该是跟地震矩等效的:

- x 绘制标准的 GCMT 解
- y 只绘制地震矩的 double couple 部分 (DC)
- z 绘制 0 迹地震矩

## 选项

**<table>** 输入文件，文件中震源机制信息，具体格式由 **-S** 选项决定。

**-C[<pen>] [P<pointsize>]** 绘制 (X,Y) 和 (newX,newY) 的连线

将震源球放在输入文件的 **newX** 和 **newY** 所指定的位置，在震源位置处绘制一个小圆，并绘制一条直线连接小圆和震源球。

**<pen>** 指定直线的属性，**<pointsize>** 指定圆的大小。

**-D<depmin>/<depmax>** 只绘制震源深度在 **<depmin>** 和 **<depmax>** 之间的地震。

**-E<fill>** 扩张部分的填充色，默认为白色

**-F<mode> [<args>]** 设置多个属性，可重复使用多次。

- **-Fa[<size>] [/<P\_axis\_symbol> [<T\_axis\_symbol>]]** 在震源球上 P 轴和 T 轴处绘制符号。  
**<size>** 是符号大小；符号可以取 **cld|hli|pls|tlx**（具体含义见 [psxy](#) 的 **-S** 选项），默认值为 **6p/cc**，即在 P 轴和 T 轴处分别绘制一个大小为 6p 的圆
- **-Fe<fill>** 设置 T 轴符号的填充色
- **-Fg<fill>** 设置 P 轴符号的填充色
- **-Fp<pen>** 设置 P 轴符号的画笔属性
- **-Ft<pen>** 设置 T 轴符号的画笔属性
- **-Fo** 使用旧版本的 **psvelomeca** 命令的输入数据格式，即不需要第三列的深度信息
- **-Fr<fill>** 在标签后加一个方框
- **-Fz<pen>** 覆盖零迹矩张量的画笔属性

**-G<fill>** 指定压缩部分的填充色，默认值为黑色

**-L<pen>** 设置震源球外部轮廓的线条属性。

**-M** 所有震级使用相同的大小，具体大小由 **-S** 选项的 **<scale>** 参数决定。

**-N** 地图区域外的震源球也要绘制，默认不绘制。

**-T<num\_of\_planes> [/<pen>]** 绘制断层平面。**<num\_of\_planes>** 可以取：

- 0 绘制两个断层面
- 1 绘制第一个断层面
- 2 绘制第二个断层面

**-W<pen>** 同时设置所有线条以及符号轮廓的画笔属性以及标题颜色。

该选项设置的属性可以被 **-C**、**-L**、**-T**、**-Fz** 指定的属性替代。

**-Z<cpt>** 指定 CPT 文件，根据数据文件中第三列的值（即地震深度）确定震源球的压缩部分的颜色。

## 示例

```
#!/bin/bash
PS=psmeca_ex1.ps

gmt pscoast -Rg -JQ10i -Bx60 -By30 -A10000 -Within -G200 -K > $PS
gmt psmeca -R -J -Sc1/12/0.2 -C0.5pP2p -Egrey -Gred -L1p -W1p -O << EOF >> $PS
-176.96 -29.25 47.8 202 30 93 18 60 88 9.56 26 -160 -10 010176A
EOF
rm gmt.*
```

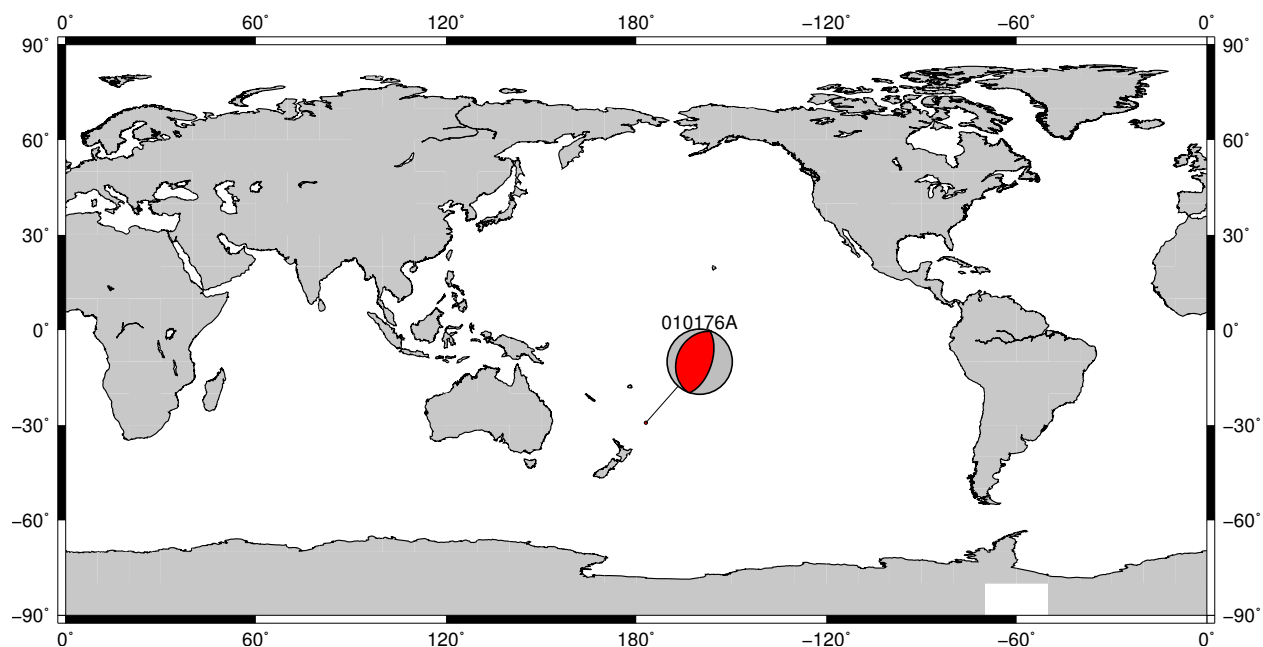


图 93.1: psmeca 示例图 1

```
#!/bin/bash
# test/meca/meca_1.sh
# Check gmt psmeca for plotting beach balls

ps=psmeca_ex2.ps

# Right lateral Strike Slip
echo 0.0 5.0 0.0 0 90 0 5 0 0 Right Strike Slip | gmt psmeca -Sa2.5c -Gblack \
  -R-1/4/0/6 -JM14c -P -B2 -K > $ps

# Left lateral Strike Slip
echo 2.0 5.0 0.0 0 90 180 5 0 0 Left Strike Slip | gmt psmeca -Sa2.5c -Gblack -R -J -K -O >> $ps
```

```

# Thrust
echo 0.0 3.0 0.0 0 45 90 5 0 0 Thrust | gmt psmeca -Sa2.5c -Gblack -R -J -K -O >> $ps
echo 2.0 3.0 0.0 45 45 90 5 0 0 Thrust | gmt psmeca -Sa2.5c -Gblack -R -J -K -O >> $ps

# Normal
echo 0.0 1.0 0.0 0 45 -90 5 0 0 Normal | gmt psmeca -Sa2.5c -Gblack -R -J -K -O >> $ps
echo 2.0 1.0 0.0 45 45 -90 5 0 0 Normal | gmt psmeca -Sa2.5c -Gblack -R -J -K -O >> $ps

# Mixed
echo 3.4 0.6 0.0 10 35 129 5 0 0 Mixed | gmt psmeca -Sa2.5c -Gblack -R -J -O >> $ps
rm gmt.*

```

```

#!/bin/bash
# test/meca/meca_2.sh

ps=psmeca_ex3.ps

gmt gmtset PROJ_LENGTH_UNIT inch MAP_TICK_LENGTH_PRIMARY 0.075i MAP_FRAME_WIDTH 0.1i \
    MAP_ORIGIN_X 2.5c MAP_ORIGIN_Y 1.3i

# This is a strike-slip CMT mechanism in red
# Best double couple overlays moment tensor
gmt psmeca -X2i -Y5i -R238.5/242/32.5/35.5 -Jm1.3i -B2 -BWeSn+tpsmeca -Sm0.4i/12pu -TO \
    -P -C0.5pP5p -Gred -K -W0.75p << EOF > $ps
# lon    lat    dep mrr mtt mff mrt mrf mtf exp plon plat text
239.384 34.556 33 -.27 -2.13 2.40 -.07 -1.32 -.79 24 240.0 35 tensor
EOF

# Focal mechanisms are labeled with following font
gmt gmtset FONT_ANNOT_PRIMARY Times-Roman
# Here are several thrusting mechanisms in Aki and Richards convention
# New second argument -Sa is font size in points
# Old gmt psvelomeca format is used

gmt psmeca -Fo -R -J -Sa0.4i/16p -C0.25p,blueP5p -O -K -N << EOF >> $ps
# lon    lat    strike dip rake Mw plon plat text
241.459 34.2088 112.3 42.2 89.8 6.6 238. 35.0 first
241.459 34.2088 120.0 60.0 86.0 5.2 238. 34.0 second
241.459 34.2088 290.0 55.0 90.0 5.9 238. 33.0 third
EOF

# Plot P and T axis and only the best double couple
# derived from moment tensor
gmt psmeca -R -J -Sd1c -O -K -N -Fa0.2c/id << EOF >> $ps

```

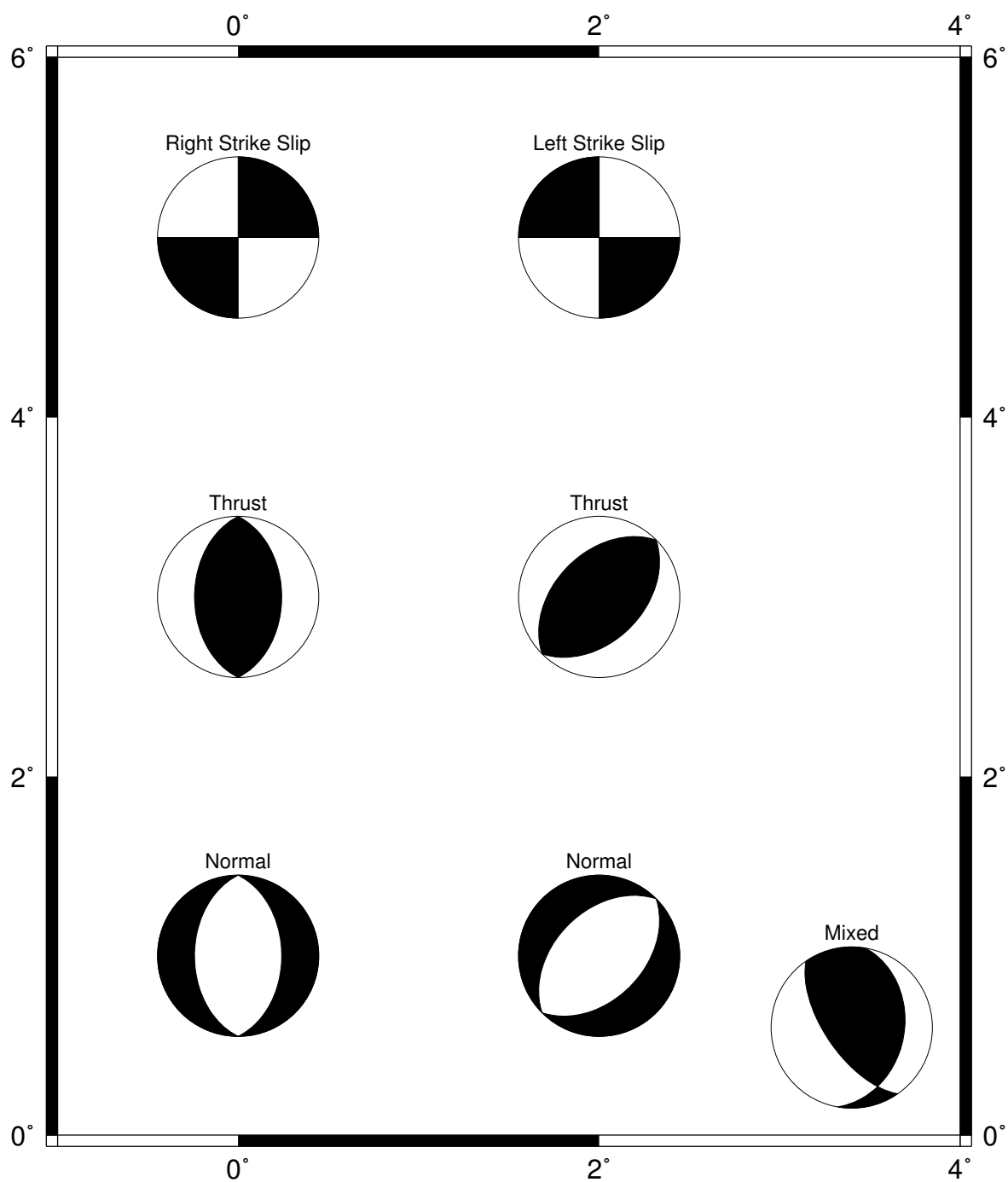


图 93.2: psmeca 示例图 2

```
# lon  lat  dep  mrr  mtt  mff  mrt  mrf  mtf  exp  plon plat
    241   33   3   -.27 -2.13 2.40 -.07 -1.32 -.79 26    0    0
EOF

# hit the beach
gmt pscoast -O -R -J -W1p -Di >> $ps
rm gmt.*
```

## psmeca

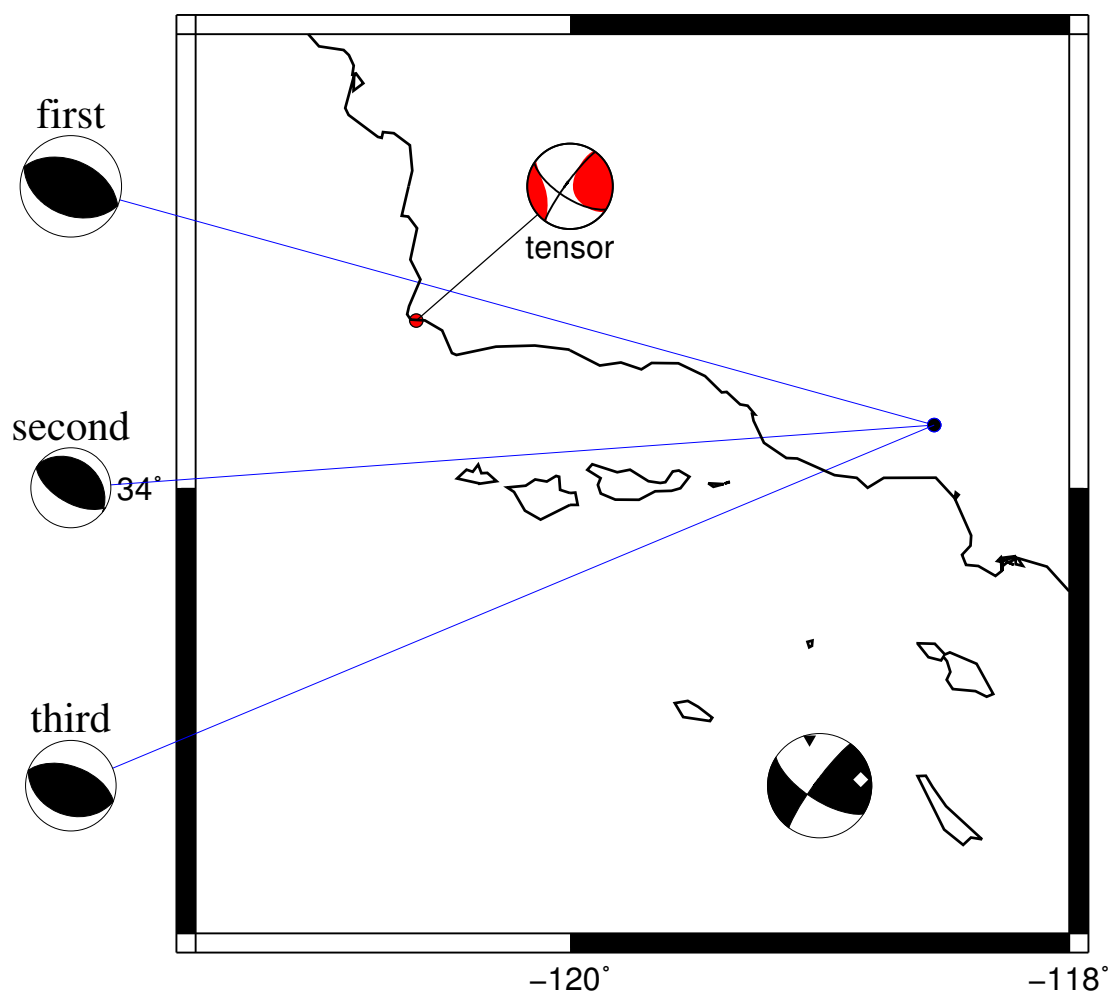


图 93.3: psmeca 示例图 3

---

## pspolar

---

官方文档 [pspolar](#)

**简介** 将台站的极性信息画在震源球上

一般情况下，需要使用 `psmeca` 绘制震源球，再使用 `pspolar` 将每个台站的极性信息画在相应的震源球上。因而，通常 `psmeca` 和 `pspolar` 需要连在一起使用。

## 选项

`<infile>` 输入数据的格式为:

```
station_code azimuth take-off_angle polarity
```

- `station_code` 通常是台站名
- `azimuth` 震相从源到台站的方位角
- `take-off_angle` 震相从源出发时的出射角
- `polarity`
  - 压缩部分（正极性）：可以取 `c|C|u|U|+`
  - 拉伸部分（负极性）：可以取 `d|D|r|R|-`
  - 未定义：其他字符

`-C<new_lon>/<new_lat>[W<pen>][P<pointsize>]` 将震源球放在新的位置，并将新位置与老位置之间连线。

`-D<lon>/<lat>` 震源球的位置，需要与 `psmeca` 输入数据中震源球的位置相同

`-E<color>` 拉伸象限内台站的符号填充色

`-F<color>` 设置震源球的背景色，默认不填充。

`-G<color>` 压缩象限内台站的符号填充色，默认值为黑色



**-M<size>** 震源球的尺寸，需要与 psmeca 中 -S 选项中震源球尺寸相同

**-S<symbol\_type><size>** 类似于 psxy 的 -S 选项，控制极性要以什么符号以及多大尺寸绘制到震源球上，可选的符号类型包括 a|c|d|h|i|p|s|t|x

**-N** 不跳过地图边界外的符号

**-Q<mode><args>** 设置多个属性，该选项可重复使用。

- -Qe[<pen>] 拉伸象限内符号的轮廓属性
- -Qf[<pen>] 震源球的轮廓属性
- -Qg[<pen>] 压缩象限内符号的轮廓属性
- -Qh 使用 HYPO71 输出的特殊格式
- -Qs<half-size>/[V[<vecpar>]][G<fill>][L] 绘制 S 波偏振方位角，见官方文档
- -Qt[<pen>] station\_code 的字体颜色

**-T<angle>/<form>/<justify>/<fontsize>** 将 station\_code 写到图上，默认值为 0.0/0/5/12。  
<fontsize> 的单位为 p

**-W<pen>** 设置画笔属性

## 示例

```
#!/bin/bash
ps=pspolar_ex1.ps

gmt gmtset PROJ_LENGTH_UNIT inch MAP_TICK_LENGTH_PRIMARY 0.075i MAP_FRAME_WIDTH 0.1i \
    MAP_ORIGIN_X 2.5c MAP_ORIGIN_Y 1.3i FONT_TITLE 18p

# this is Harvard CMT for tibethan earthquake (1997)
gmt psmeca -Fo -R85/89/25/50 -JX7i -P -M -Sm4i -N -L -K -G150 -T0 << EOF > $ps
# lon lat mrr mtt mff mrt mrf mtf ex nlon nlat
87 35 -0.26 -0.71 0.97 -0.20 -0.61 2.60 27 0 0
EOF

# and polarities observed
gmt pspolar -R -J -D87/35 -M4i -N -Sc0.3i -Qe -0 \
    -B0 -B+t"Tibet earthquake (1997) - polarities distribution" << EOF >> $ps
1 147.8 53 c
2 318.6 53 c
3 311.9 53 c
4 122.5 45 c
5 87.1 44 c
6 259.9 44 c
```

7 358.0 43 d  
8 32.3 40 d  
9 144.5 40 c  
10 206.2 40 d  
11 30.0 36 d  
12 88.3 31 c  
13 326.5 31 c  
14 298.4 29 c  
15 298.3 29 c  
16 316.2 28 c  
17 301.5 27 c  
18 300.7 27 c  
19 303.0 27 d  
20 302.7 26 c  
21 301.7 26 c  
22 302.3 26 c  
23 302.2 26 c  
24 314.1 26 c  
25 296.2 26 c  
26 302.3 26 c  
27 146.8 26 c  
28 145.7 26 d  
29 145.7 26 c  
30 307.0 26 c  
31 311.9 26 c  
32 136.4 25 c  
33 297.6 25 c  
34 306.1 25 c  
35 306.8 25 c  
36 307.6 25 c  
37 346.5 25 c  
39 306.5 24 c  
40 317.3 24 c  
41 305.2 24 c  
42 305.9 24 c  
43 311.9 24 c  
44 307.5 24 c  
45 138.7 24 d  
46 322.4 24 c  
47 305.3 24 c  
48 304.9 24 c  
49 309.3 24 c  
50 307.6 24 c  
51 315.5 24 d  
52 310.3 24 c

```
53 308.5 24 c
54 307.4 24 c
55 307.5 24 c
56 307.4 24 c
57 307.6 24 c
58 307.1 24 c
59 311.5 23 d
61 243.5 23 d
63 345.2 23 c
64 117.0 21 d
65 133.1 20 c
66 116.0 20 c
67 231.3 17 d
68 139.9 16 c
69 131.7 15 d
70 114.1 15 c
EOF
rm gmt.*
```

## Tibet earthquake (1997) – polarities distribution

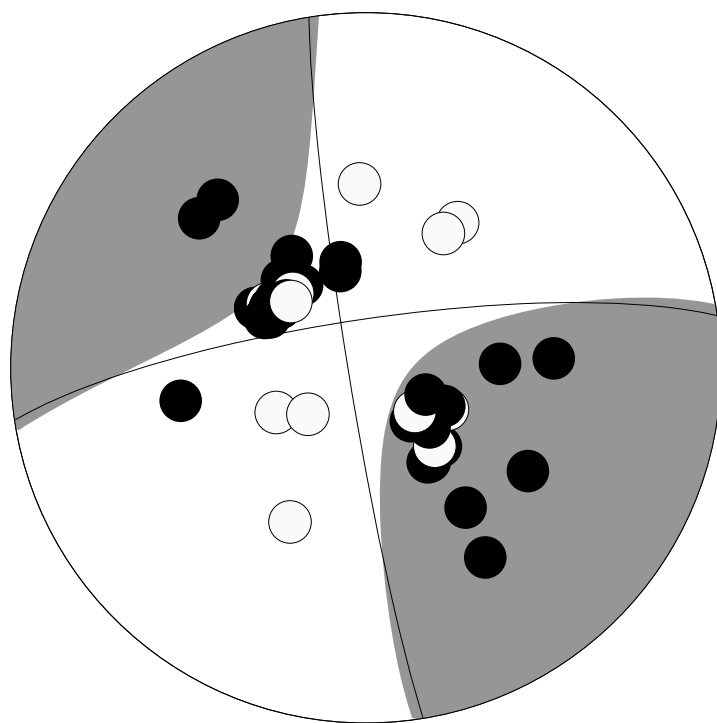


图 94.1: pspolar 示例

官方文档 [psrose](#)

简介 绘制极坐标下的直方图 (sector 图、rose 图或 windrose 图)

## 选项

**<table>** 输入文件，数据格式为 `length azimuth`

若输入文件中只有 `azimuth` 一列数据，则此时需要使用 `-i` 选项，此时所有的长度都默认为单位长度。

**-A[r]<sector\_width>** 指定扇页宽度，单位为度

1. 默认扇页宽度为 0，即 windrose 图
2. 若扇页宽度不为 0，则表示绘制 sector 图
3. 若扇页宽度不为 0，则使用了 `r`，则表示绘制 rose 图

**-B** 见 `-B` 选项的介绍。此处，`X` 表示径向距离，`Y` 表示方位角。`Y` 轴的标签是图片的标题。

**-C[mode\_file]** 绘制矢量以显示 `<mode_file>` 中指定的主方向。若未指定 `<mode_file>` 则计算并绘制平均方向。

**-D** 对扇页对偏移，使得其位于每个间隔的中间，即第一个扇页的中心在 0 度处

**-F** 不绘制 scale length bar，默认会在右下角绘制

**-G<fill>** 绘制扇页的填充色

**-I** 不绘制图形，计算 `-R` 选项所需要的参数

**-L[<wlabel>/<elabel>/<slabel>/<nlabel>]** 指定 0、90、180、270 度处的标签。

1. 对于 full-circle 图而言，默认值为 WEST/EAST/SOUTH/NORTH
2. 对于 half-circle 图而言，默认值为 90W/90E/-/0，其中 `-` 表示不显示标签
3. 只使用 `-L` 但无其他参数表示不显示所有标签

**-M<parameters>** 与 **-C** 选项一起使用以修改矢量的属性。具体数据见 [vectors](#) 一节

**-R<r0>/<r1>/<az\_0>/<az\_1>** 指定绘图的半径和方位角范围。

**-S[n]<radial\_scale>** 指定圆的半径。

**-Sn** 会将输入的半径归一化到 0 到 1。

**-T** Specifies that the input data is orientation data (has a 180 degree ambiguity) instead of true 0-360 degree directions [Default].

**-W[v]<pen>** 设置扇区边框的画笔属性。

**-Wv<pen>** 可用于设置绘制矢量时所需的画笔属性。

**-Z<scale>** 将数据的半径乘以 **<scale>**

比如 **-Z0.001** 会将数据的单位从 m 变成 km。若不考虑半径，可以通过 **-Zu** 将所有的半径设置为单位长度。

## psscale

官方文档 [psscale](#)

简介 在地图上绘制灰色或彩色色条 (colorbar)

## 选项

**-B[p|s]<parameters>** 设置 colorbar 的标注、刻度和网格线间隔。

在不使用 **-B** 选项，或者不指定标注间隔时，默认会根据 CPT 文件中每一行的内容对 colorbar 进行标注，具体见 [cpt](#)。

默认情况下，对于水平 colorbar 而言，X 轴的标签会放在 colorbar 的下边，Y 轴标签放在 colorbar 的右边；对于垂直 colorbar 而言，X 轴的标签放在 colorbar 的右边，Y 轴标签放在 colorbar 的上边。

**-C** 要绘制的 CPT 文件。

**-D[g|j|J|n|x]<refpoint>+w<length>/<width>[+e[b|f][<length>]][+h][+j<justify>][+m[a|c|l|u]][+n[<text>]][+<fill>]**  
指定色标的尺寸和位置。

1. **[g|j|J|n|x]<refpoint>** 指定底图上的参考量，见 [embellishments](#) 一节
2. **+j<justify>** 指定色标上的锚点，默认锚点是 BL，见 [embellishments](#) 一节
3. **+o<dx>[/<dy>]** 指定参考点的额外偏移量，见 [embellishments](#)
4. **+w<length>/<width>** 指定色标的长度和宽度，若长度为负值则会反转色标
5. **+h** 绘制水平色标，默认为垂直色标
6. **+e** 选项为前景色和背景色加一个三角形，**+ef** 表示只加前景色三角形，**+eb** 表示只加背景色三角形，**<length>** 是三角的高度，默认为色标宽度的一半
7. **+m[a|c|l|u]** 将标注、标签和单位放在色标的另一边，**a** 代表标注，**l** 代表标签，**u** 代表单位。**c** 表示将标签以单列字符垂直打印。
8. **+n<text>** 在色标开始处绘制一个矩形，并用 NaN 的颜色填充

-F 为 colorbar 加上背景边框，见 [embellishments](#) 一节

-G<zlow>/<zhigh> 绘图前先对 CPT 文件做截断，使得其只绘制 <zlow> 到 <zhigh> 之间的部分。若其中某个值等于 NaN，则不对 CPT 的那一端做处理。

-I[<max\_intens>|<low\_i>/<high\_i>] 为色标加上光照效果

1. -I<max\_intens> 光照强度为 [-<max\_intens>,<max\_intens>]，默认值为 1

2. -I<low\_i>/<high\_i> 指定非对称的光照强度范围

-L[i][<gap>] 生成等大小的 color rectangles，见官方文档

-M 使用 YIQ 变换将色标变成单调灰度色标

-N[p|<dpi>] 见官方文档

-Q 使用对数坐标，刻度表示为 10 的次幂

CPT 文件中所有的 Z 值都会被转换成  $p = \log_{10}(z)$ ，其中整数的 p 会以  $10^p$  的格式标注。

-S 去除不同色块之间的黑色网格线

-Z<zfile> 见官方文档

## 示例

```
gmt makecpt -T-200/1000/100 -Crainbow > t.cpt
gmt psscale -Ct.cpt -Dx8c/1c+w12c/0.5c+jTC+h -Bxaf+l"topography" -By+1km > map.ps
```



---

**pssegy**

---

官方文档 [pssegy](#)

---

**pssegyz**

---

官方文档 [pssegyz](#)

---

**pstext**

---

**官方文档** [pstext](#)**简介** 在图上写文本

该命令用于在图上添加文本。可以自由控制文本的大小、颜色、字体、角度。

通过使用转义字符、特殊字体以及八进制码，还可以实现希腊字母、上下标等功能，详见 [text](#)、[character-escape](#)、[special-fonts](#) 和 [special-characters](#)。

**选项**

**<textfiles>** 默认情况下，输入数据的格式为：

```
X Y text
```

如果使用了 **-F** 选项，则输入数据的格式为：

```
x y [font angle justify] text
```

其中，括号内的三项是否存在以及三项的顺序由 **-F** 选项决定。

**-A** 默认情况下，**angel** 是指沿水平方向逆时针旋转的角度。**-A** 选项表明 **angle** 是方位角，即相对于北向顺时针旋转的角度。

**-C<dx>/<dy>** 设置文本框与文本之间的空白区域，默认值为字体大小的 15%。

**<dx>** 可以是具体的距离值也可以接 % 表示空白与当前字号的百分比。例如 **-C1c/1c** 或 **-C20%/30%**。

下图展示了文本与文本框之间的空隙。

**-D[j|J]<dx>/<dy>[v<pen>]]** 文本在指定坐标的基础上偏移 **<dx>/<dy>**，默认值为 0/0，即不偏移。

使用 **pstext** 经常遇到的情况是在台站处标记台站名，此时传递给 **pstext** 的位置参数通常是台站坐标，因而 **pstext** 会将文本置于台站坐标处，该选择用于将文本稍稍偏离台站坐标位置以避免文本挡住台站处的符号。



1. 若不指定 `<dy>`，则默认 `<dy>=<dx>`
2. `-Dj` 见官方文档
3. `-DJ` 见官方文档
4. 偏移量后加上 `v` 表示绘制一条连接初始位置与偏移后位置的直线
5. `v<pen>` 控制连线的画笔属性

`-F[+a[angle]][+c[justify]][+f[font]][+h][+j[justify]][+l]` 控制文本的角度、对齐方式和字体等属性。

1. `+f<font>` 设置文本的字体，见 [text](#)
2. `+a<angle>` 文本相对于水平方向逆时针旋转的角度
3. `+j<justify>` 文本对齐方式，见 [anchors](#)

下面的命令中，统一设置了所有文本的字号为 30p，4 号字体，红色，文本旋转 45 度，且以左上角对齐：

```
gmt pstext -R0/10/0/10 -JX10c/10c -B1g1 -F+f30p,4,red+a45+jTL > text.ps << EOF
3 4 Text1
6 8 Text2
EOF
```

若使用了 `+f` 子选项，但是未给定 `<font>`，则意味着输入数据的每一行需要自定义本行的字体属性，因为输入数据的格式要发生变化。例如 `-F+f` 选项要求的输入数据的格式为：

```
x y font text
```

对于 `+a` 和 `+j` 同理。若 `+f`、`+a`、`+j` 中有两个以上未在命令行中指定参数，则输入数据中要增加多列，每列的顺序由这三个子选项的相对顺序决定。比如 `-F+f+a` 的输入数据格式是：

```
x y font angle text
```

`-F+a+f` 的输入数据格式为：

```
x y angle font text
```

`-F+f+j+a` 表示所有数据都需要单独指定字体、对齐方式和角度，此时输入数据的格式为：

```
x y font justification angle text
```

-F+a+j+f 与前一个例子类似，唯一的区别在于子选项的顺序不同，而输入数据的格式要与子选项的顺序相匹配，此时输入数据的格式为：

```
x y angle justification font text
```

-F+f12p,Helvetica-Bold,red+j+a 为所有行设置了统一的字体，但每一行需要单独指定对齐方式和角度，此时输入数据的格式为：

```
x y justification angle text
```

使用 +c<justify> 选项，则输入数据中不需要 XY 坐标，只需要文本即可，该选项直接从 -R 选项中提取出范围信息，并由对齐方式决定文本的坐标位置。比如 -F+cTL 表示将文本放在底图的左上角，在加上合适的偏移量即可放在任意位置。例如：

```
echo '(a)' | gmt pstext -R0/10/0/10 -JX10c/10c -B1 -F+cTL -Dj0.2c/0.2c > text.ps
```

+h 会直接从多段数据的段头记录中提取文本：

```
gmt pstext -R0/10/0/10 -JX10c/10c -B1 -F+h > text.ps << EOF
> TEXT1
2 2
> TEXT2
5 5
EOF
```

+l 会直接从多段数据的段头记录里的 -L<label> 中提取信息：

```
gmt pstext -R0/10/0/10 -JX10c/10c -B1 -F+h > text.ps << EOF
> -LTEXT1
2 2
> -LTEXT2
5 5
EOF
```

-G 设置文本框的填充色。

除了设置填充色之外，-G 选项还有两个高级用法，即 -Gc 和 -GC 。其中，-Gc 表示先绘制文本，然后将文本框裁剪出来，并打开裁剪选项，之后的绘图命令都不会覆盖文本所在区域，最后需要使用 *psclip* 的 -C 选项关闭裁剪。若不需要绘制文本只想要激活裁剪选项，可以使用 -GC 选项。

-L 用于列出 GMT 所支持的所有字体名及其对应的字号：

```
gmt pstext -L
```

**-M** 段落模式，用于输入大量文本。

输入文件必须是多段数据。数据段头记录的格式为:

```
> X Y [font angle justify] linespace parwidth parjust
```

1. 第一个字符是数据段开始标识符，默认为 >
2. 从第三列开始，包含了本段文本的设置信息
3. font angle justify 是可选的，由 -F 选项控制
4. linespace 行间距
5. parwidth 段落宽度
6. parjust 段落对齐方式，可以取为 l（左对齐）、c（居中对齐）、r（右对齐）、j（分散对齐）

段头记录后即为要显示在图上的文本，每段数据之间用空行分隔。

```
#!/bin/bash
gmt pstext -R0/3/0/5 -JX3i -F+f+a+j -B1 -M -N -Y4c << EOF > pstext_-M.ps
> 0 -0.5 12p,black 0 LT 13p 3i j
@%5%Figure 1.%% This illustration shows nothing useful, but it still needs
a figure caption. Highlighted in @;255/0/0;red@;; you can see the locations
of cities where it is @_impossible@_ to get any good Thai food; these are to be avoided.
EOF

rm gmt.*
```

**-N** 位于地图边界外的文本也被绘制。

默认情况下，若文本超过了底图边框，则不显示该文本，即文本被裁剪掉了。使用 -N 选项，即便文本超出了底图边框的范围，也依然会显示。

**-Ql|u** 所有文本以小写（lower case）或大写（upper case）显示

**-To|O|c|C** 设置文本框的形状

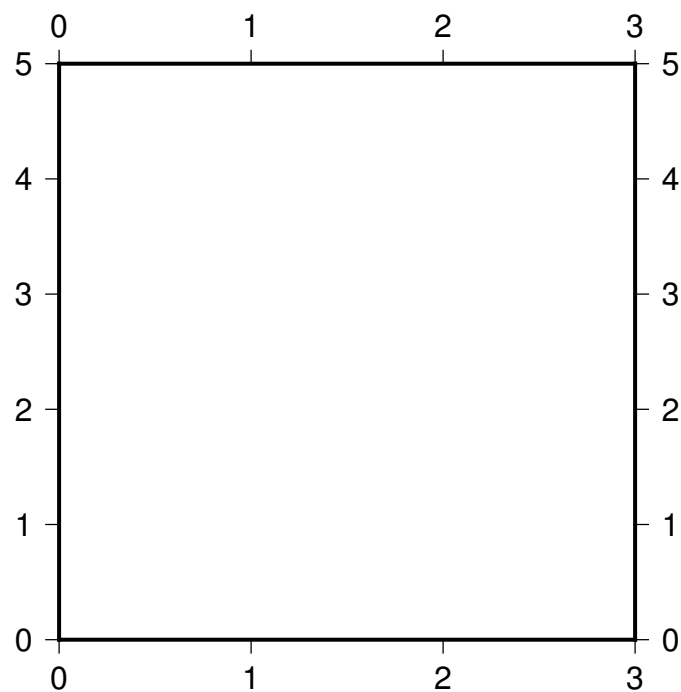
1. -To : 直角矩形
2. -TO : 圆角矩形
3. -Tc : 凹矩形（与 -M 选项一起使用）
4. -TC : 凸矩形（与 -M 选项一起使用）

**-W<pen>** 设置文本框的边框属性，默认值为 default,black,solid

**-Z** 3D 投影中，需要在数据的第三列指定文本的 Z 位置，数据格式为:

```
X Y Z Text
```

此时强制使用 -N 选项。



**Figure 1.** This illustration shows nothing useful, but it still needs a figure caption. Highlighted in **red** you can see the locations of cities where it is impossible to get any good Thai food; these are to be avoided.

图 99.1: 段落模式示意图

## 示例

下面的例子中设置文本框的相关属性：蓝色边框、淡蓝填充色、圆角矩形，空白为 100%/100%

```
gmt pstext -R0/10/0/5 -JX10c/5c -B1 -Wblue -Glightblue -T0 -C100%/100% > text.ps << EOF
3   1   Text1
6   3   Text2
EOF
```



官方文档 [psvelo](#)

简介 在地图上绘制速度矢量、十字线、楔形图

## -S 选项

-S 选项决定了要绘制的符号的类型，以及输入数据的格式。

-Se<velscale>/<confidence>/<fontsize> 绘制速度椭圆。速度用矢量表示，速度的不确定度用椭圆表示。输入数据中每列的含义为：

1. 台站经度
2. 台站纬度
3. 速度矢量的东向分量
4. 速度矢量的北向分量
5. 东向分量的不确定度 (1-sigma)
6. 北向分量的不确定度 (1-sigma)
7. 东向分量和北向分量的 correlation
8. 台站名 (可选)

说明：

1. <velscale> 控制了速度矢量的缩放比例，其对应了速度值为 1 的箭头的长度，默认单位由参数 PROJ\_LENGTH\_UNIT 决定，也可以自己添加长度单位 `ilclp`
2. <confidence> 设置了椭圆的 2D 置信区间的极限，比如 0.95 代表 95% 置信区间
3. <fontsize> 是文本（台站名）的大小
4. 椭圆的填充色由 -G 选项控制

5. 椭圆的轮廓以及矢量的画笔属性由 -W 选项控

**-Sr<velscale>/<confidence>/<fontsize>** 绘制速度椭圆，与 -Se 类似，区别在于输入文件不同：

1. 经度
2. 纬度
3. 速度矢量的东向分量
4. 速度矢量的北向分量
5. 椭圆的半长轴
6. 椭圆的半短轴
7. 椭圆的长轴相对于水平方向逆时针旋转的角度
8. 台站名（可选）

**-Sn<barscale>** 绘制各向异性矢量，<barscale> 是各向异性矢量的缩放比例。输入数据中每列的含义为：

1. 经度
2. 纬度
3. 各向异性矢量的东向分量
4. 各向异性矢量的北向分量

**-Sw<wedge\_scale>/<wedge\_mag>** 绘制旋转饼图。输入数据中每列的含义为：

1. 经度
2. 纬度
3. 旋转的角度（以 radians 为单位）
4. 旋转的不确定度（以 radians 为单位）

<wedge\_scale> 设置了饼图的大小。所有的数值在绘图之前会先乘以 <wedge\_mag> 。饼图的填充色由 -G 选项控制，不确定度的颜色由 -E 控制。

**-Sx<cross\_scale>** 绘制应变十字线。<cross\_scale> 控制十字线的大小。输入文件中每列的含义为：

1. 经度
2. 纬度
3. eps1: 应变张量的扩张本征值，扩张为正
4. eps2: 应变张量的压缩本征值，扩张为正
5. eps2 的方位角（北方向顺时针旋转的角度）

## 选项

**-A<params>** 修改矢量的属性，见 [vectors](#) 一节

**-D<sigma\_scale>** 控制 **-Se** 和 **-Sr** 中速度不确定度以及 **-Sw** 中旋转不确定度的缩放。该参数可以与 **<confidence>** 一起控制不确定度的绘制

**-F<fill>** 控制边框和标注的颜色

**-E<fill>** 控制 **-Sw** 中饼图不确定度的填充

**-G<fill>** 控制颜色？

**-L** 绘制线段。使用该选择，则椭圆和断层面会根据 **-W** 选项绘制轮廓

**-N** 不跳过区域范围外的符号

**-W<pen>** 设置速度箭头、椭圆轮廓以及断层边界的颜色

## 示例

```
#!/bin/bash
ps=psvelo_ex1.ps

# 左边三个符号
gmt psvelo << EOF -Xc -JM15c -R241/242/34/35 -B0.5 -BWeSn+tpsvelo \
    -Sw3c/1.e7 -Wlp -G60 -E210 -D2 -P -K > $ps
# lon    lat    spin(rad/yr) spin_sigma (rad/yr)
241.2 34.2  5.65E-08 1.17E-08
241.2 34.5 -4.85E-08 1.85E-08
241.2 34.7  4.46E-09 3.07E-08
EOF

# 右边三个符号
gmt psvelo -J -R -Sw3c/1.e7 -Wlp -D2 -O -K << EOF >> $ps
# lon    lat    spin(rad/yr) spin_sigma (rad/yr)
241.7 34.2  5.65E-08 1.17E-08
241.7 34.5 -4.85E-08 1.85E-08
241.7 34.7  4.46E-09 3.07E-08
EOF
gmt pscoast -O -R -J -Wlp -Di >> $ps
rm gmt.*
```

```
#!/bin/bash
ps=psvelo_ex2.ps
```

# psvelo

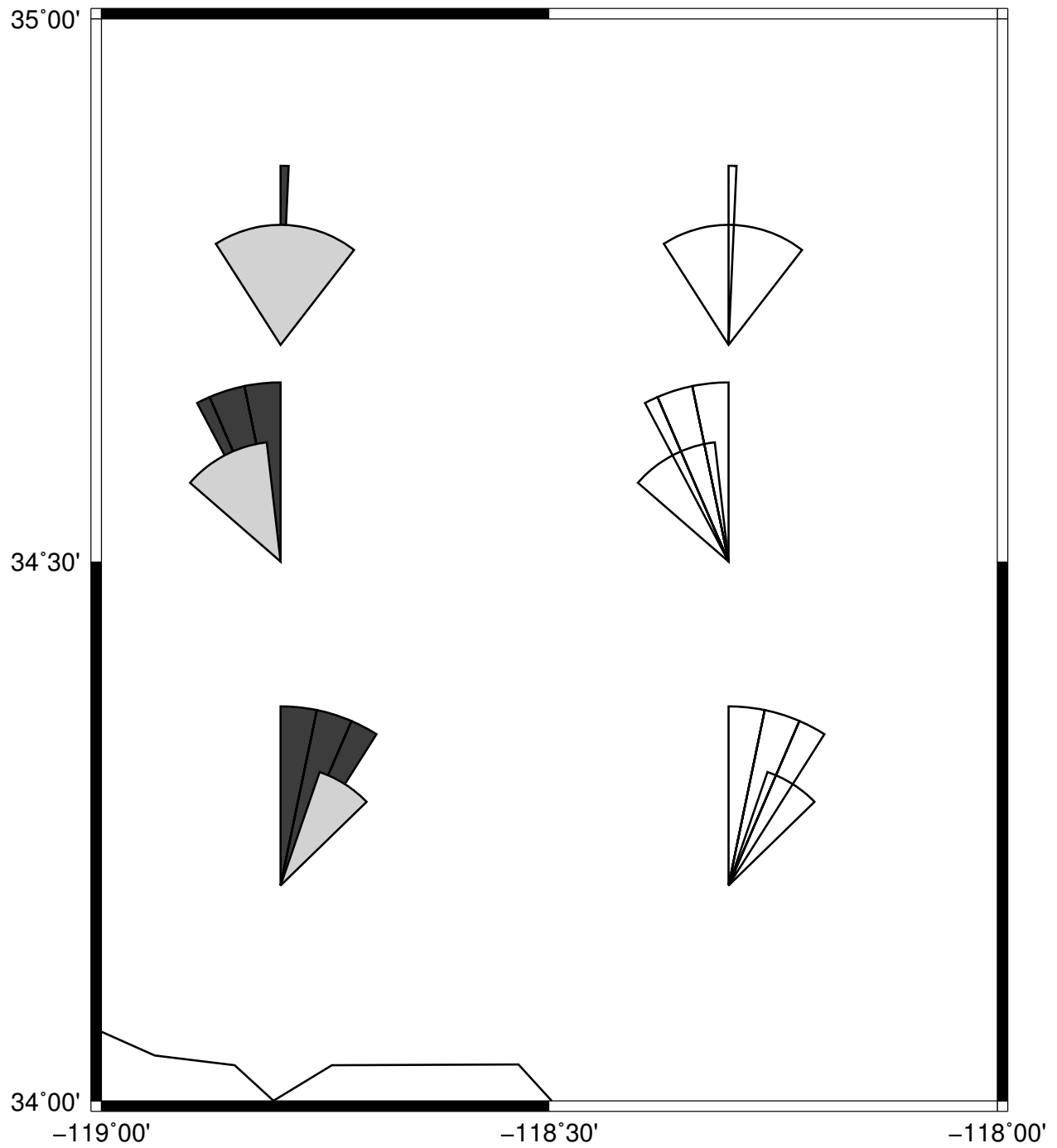


图 100.1: psvelo 示例图 1

```
gmt psvelo -R-10/10/-10/10 -Wthin,red \
  -Se0.2c/0.39/12 -B1g1 -BWeSn -Jx0.2i -Ggreen -Eblue -L -N \
  -A1c+p3p+e -K << EOF > $ps
# Long.   Lat.   Evel   Nvel   Esig   Nsig   CorEN SITE
# (deg)   (deg)   (mm/yr)   (mm/yr)
  -8.     0.     5.0    0.0    4.0    6.0    0.500 4x6
  -5.     0.     0.0    5.0    4.0    6.0    0.500 4x6
  -5.     5.    -5.0    0.0    4.0    6.0    0.500 4x6
  -8.     5.     0.0   -5.0    0.0    0.0    0.500 4x6
  -1.     5.     3.0    3.0    1.0    1.0    0.100 3x3
EOF

gmt psvelo -Se0.2c/0.39/14 -R -J -A0.25c+p0.25p+e -O << EOF >> $ps
# Long.   Lat.   Evel   Nvel   Esig   Nsig   CorEN SITE
# (deg)   (deg)   (mm/yr)   (mm/yr)
   0.    -8.     0.0    0.0    4.0    6.0    0.100 4x6
  -8.     5.     3.0    3.0    0.0    0.0    0.200 3x3
   0.     0.     4.0    6.0    4.0    6.0    0.300
  -5.    -5.     6.0    4.0    6.0    4.0    0.400 6x4
   5.     0.    -6.0    4.0    6.0    4.0   -0.300 -6x4
   0.    -5.     6.0   -4.0    6.0    4.0   -0.500 6x-4
EOF

rm gmt.*
```

下面的示例展示了如何绘制应变十字线:

```
gmt psvelo -JX10c/10c -R0/10/0/10 -Sx10 -W1p > test.ps << EOF
5 5 0.5 0.3 45
EOF
```

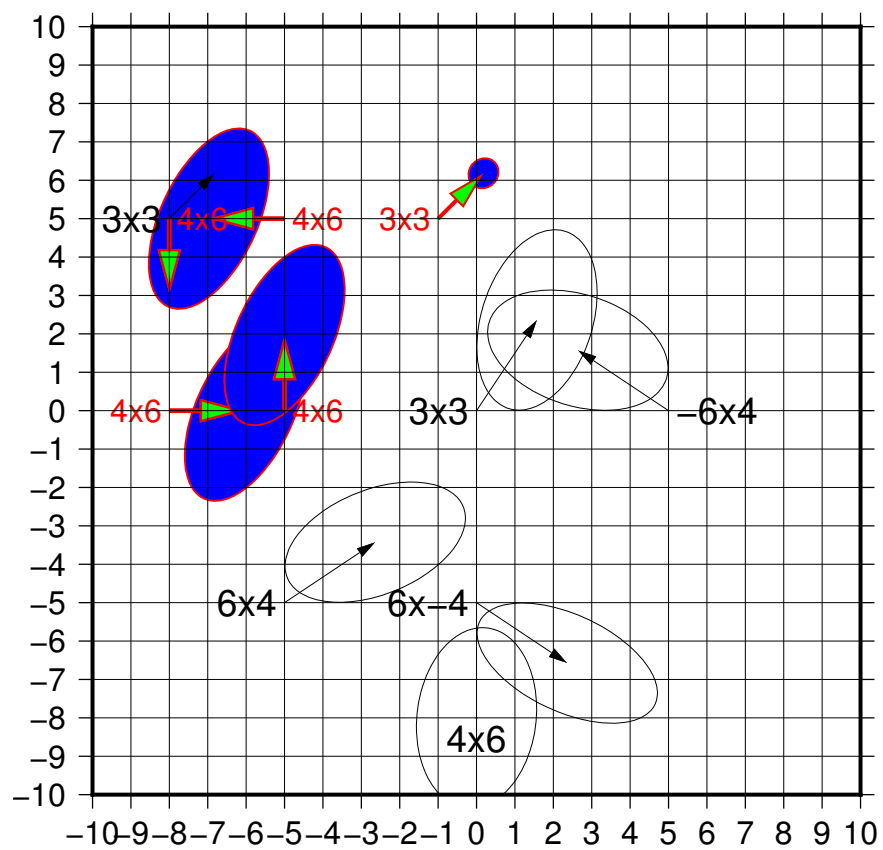


图 100.2: psvelo 示例图 2

---

**pswiggles**

---

官方文档 [pswiggles](#)

### 官方文档 [psxy](#)

**简介** 在图上绘制线段、多边形和符号

该命令既可以用于画线段（多边形可以认为是闭合的线段）也可以用于画符号，唯一的区别在于是否使用了 `-S` 选项。在不使用 `-S` 选项的情况下，默认会将所有的数据点连成线，使用 `-S` 选项则仅在数据点所在位置绘制符号。

## 选项

`-A[m|p|x|y]` 修改两点间的连接方式。

地理投影下，两点之间默认沿着大圆弧连接。

1. `-A` : 忽略当前的投影方式，直接用直线连接两点
2. `-Am` : 先沿着经线画，再沿着纬线画
3. `-Ap` : 先沿着纬线画，再沿着经线画

笛卡尔坐标下，两点之间默认用直线连接。

1. `-Ax` 先沿着 X 轴画，再沿着 Y 轴画
2. `-Ay` 先沿着 Y 轴画，再沿着 X 轴画

下图中，黑色曲线为默认情况；红线为使用 `-A` 的效果；蓝线为使用 `-Ap` 的效果；黄线为使用 `-Am` 的效果：

注：由于这里投影比较特别，所以沿着经线的线和沿着纬线的线，看上去都是直线，在其他投影方式下可能不会是这样。

该选项的一个可能的用途见示例：<http://seisman.info/plot-specified-gridline-on-maps.html>

`-C<cpt>` 指定 CPT 文件或颜色列表



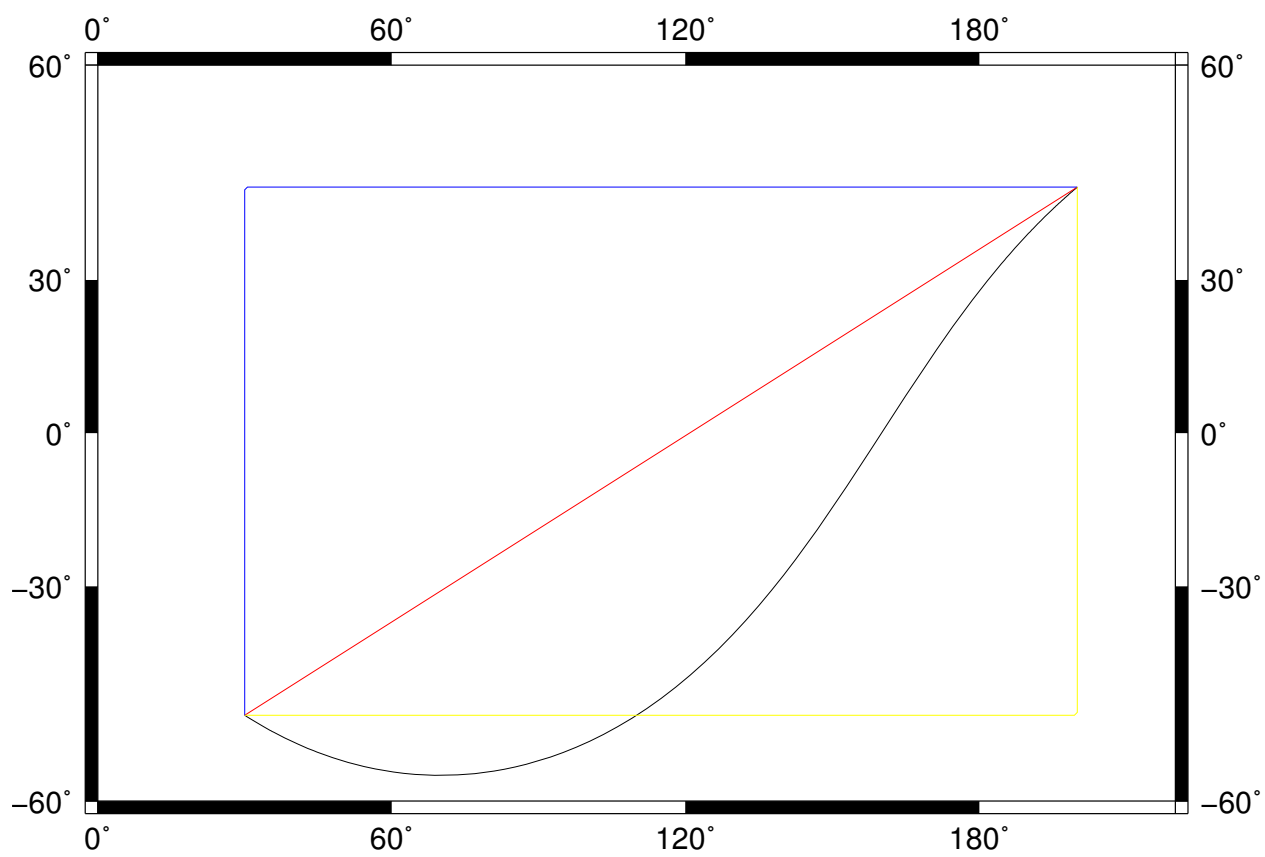


图 102.1: psxy -A 选项示意图

该选项后跟一个 CPT 文件名，也可以使用 `-C<color1>,<color2>,...` 语法在命令行上临时构建一个颜色列表，其中 `<color1>` 对应 Z 值为 0 的颜色，`<color2>` 对应 Z 值为 1 的颜色，依次类推。

1. 若绘制符号（即使用 `-S` 选项），则符号的填充色由数据的第三列 Z 值决定，其他数据列依次后移一列
2. 若绘制线段或多边形（即未使用 `-S` 选项），则需要有多段数据的头段中指定 `-Z<val>`，然后从 cpt 文件中查找 `<val>` 所对应的颜色，以控制线段或多边形的线条颜色

下面的例子测试了 `-C<color1>,<color2>..` 用法：

```
gmt psxy -JX10c/10c -R0/10/0/10 -B1 -Cblue,red -W2p > test.ps << EOF
> -Z0
1 1
2 2
> -Z1
3 3
4 4
EOF
```

`-D<dx>/<dy>` 设置符号的偏移量。

该选项会将要绘制的符号或线段在给定坐标的基础上偏移 `<dx>/<dy>` 距离。若未指定 `<dy>`，则默认 `dy=dx`。

`-E[x[+] | y[+] | X|Y] [n] [<cap>] [/[-|+]<pen>]` 绘制误差棒。

先解释几个参数：

1. `<cap>` 控制误差棒顶端帽子的长度，默认值为 7p
2. `<pen>` 控制误差棒的画笔属性，默认值为 `defalut,black,solid`
  - (a) `+<pen>` 使用 CPT 文件中的颜色作为符号填充色和误差棒画笔属性
  - (b) `-<pen>` 设置误差棒画笔属性，并关闭符号填充色

默认会绘制 X 和 Y 两个方向的误差棒。`x|y` 表示只绘制 X 方向和/或 Y 方向的误差棒，此时输入数据的格式为（具体格式由选项决定）：

X	Y	[size]	[X_error]	[Y_error]	[others]
---	---	--------	-----------	-----------	----------

例如，X 方向误差为 1:

```
echo 5 5 1 | gmt psxy -R0/10/0/10 -JX10c/10c -B1 -Sc0.1c -Ex -W2p > test.ps
```

X 方向误差为 1，Y 方向误差为 0.5:

```
echo 5 5 1 0.5 | gmt psxy -R0/10/0/10 -JX10c/10c -B1 -Sc0.1c -Exy -W2p > test.ps
```

使用 `x+|y+` 表明 X 方向和/或 Y 方向为非对称误差棒，此时输入数据的格式为：

```
X Y [size] [X_left_error X_right_error] [Y_left_error Y_right_error] [others]
```

例如:

```
echo 5 5 1 0.4 0.5 0.25 | gmt psxy -R0/10/0/10 -JX10c/10c -B1 -Sc0.1c -Ex+y+ -W2p > test.ps
```

使用 X 和 Y 则绘制 box-and-whisker (即 stem-and-leaf) 符号。以 -EX 为例, 此时数据格式为:

```
X 中位数 Y 0% 位数 25% 位数 75% 位数 100% 位数
```

25 到 75 之间的方框内可以用 -G 选项填充颜色:

```
echo 5 5 4 4.25 5.4 7 | gmt psxy -R0/10/0/10 -JX10c/10c -B1 -Sc0.1c -EX -Gred -W2p > test.ps
```

若使用 -EXY, 则输入数据中至少需要 10 列; 若在 X 或 Y 后加上了 n, 则需要在额外的第 5 列数据指定中位数的不确定性。

**-F[c|n|r][a|f|s|r|<refpoint>]** 修改数据点的分组和连接方式。

数据的分组方式有三种:

1. **a** 忽略所有数据段头记录, 即将所有文件内的所有数据点作为一个单独的组, 并将第一个文件的第一个数据点作为该组的参考点
2. **f** 将每个文件内的所有点分在一个组, 并将每一组内的第一个点作为该组的参考点
3. **s** 每段数据内的点作为一组, 并将每段数据的第一个点作为该组的参考点
4. **r** 每段数据内的点作为一组, 并将每段数据的第一个点作为该组的参考点, 每次连线后将前一个点作为新的参考点, 该选项仅与 -Fr 连用 (似乎与 -Fcs 等效?)
5. **<refpoint>** 指定某个点为所有组共同的参考点

在确定分组后, 还可以额外定义组内各点的连接方式:

- **c** 将组内的点连接成连续的线段
- **r** 将组内的所有点与组内的参考点连线
- **n** 将每个组内的所有点互相连线

在不使用 -F 选项的情况下, 默认值为 -Fcs。该选项的具体示例在后面给出。

**-G<fill>** 设置符号或多边形的填充色。多段数据中数据段头记录中的 -G 选项会覆盖命令行中的设置。

**-I<intens>** 模拟光照效果

<intens> 的取值范围为 -1 到 1, 用于对填充色做微调以模拟光照效果。正值表示亮色, 负值表示暗色, 零表示原色。

**-L[+b|d|D][+x1|r|x0][+y1|r|y0][+p<pen>]** 构建闭合多边形。

默认情况下，`psxy` 只将数据点连起来，若首尾两个点不相同，则不会形成闭合多边形。使用 `-F`，则自动将数据的首尾两个点连起来，形成闭合多边形。

除了简单的首尾相连之外，还可以给线段加上包络线（类似于线段的误差）：

1. `+d` build symmetrical envelope around  $y(x)$  using deviations  $dy(x)$  given in extra column 3
2. `+D` build asymmetrical envelope around  $y(x)$  using deviations  $dy1(x)$  and  $dy2(x)$  from extra columns 3-4.
3. `+b` build asymmetrical envelope around  $y(x)$  using bounds  $yl(x)$  and  $yh(x)$  from extra columns 3-4.
4. `+xl|r|<x0>` connect first and last point to anchor points at either  $xmin$ ,  $xmax$ , or  $x0$
5. `+yb|t|<y0>` connect first and last point to anchor points at either  $ymin$ ,  $ymax$ , or  $y0$ .

Polygon may be painted (`-G`) and optionally outlined by adding `+ppen` [no outline].

`-N[r|c]` 区域范围外的符号不会被裁剪，而会被正常绘制。

默认情况下，位于 `-R` 范围外的符号不会被绘制的。使用该选项使得即便符号的坐标位于 `-R` 指定的范围外，也会被绘制。需要注意的是，该选项对线段或多边形无效，线段和多边形总会被区域的范围裁剪。

对于存在周期性的地图而言，若符号出现在重复边界上，则会被重复绘制两次。比如：

```
gmt psxy -R0/360/-60/60 -JM10c -Bx60 -By15 -Sc2c > test.ps << EOF
360 0
EOF
```

会在地图的左右边界处分别两个半圆，该行为可以通过 `-N` 选项修改：

1. `-N` 关闭裁剪，符号仅绘制一次
2. `-Nr` 关闭裁剪，但符号依然绘制两次
3. `-Nc` 不关闭裁剪，但符号仅绘制一次

`-T` 忽略所有输入文件，包括标准输入流

该选项会忽略命令行中的输入文件以及标准输入流，在 Linux 下相当于将空文件 `/dev/null` 作为输入文件，因而该命令不会在 PS 文件中绘制任何图形。

该选项有如下几个用途：

1. `psxy -J$J -R$R -T -K > $PS` 只写入文件头，见 `-K` 和 `-O` 选项的介绍
2. `psxy -J$J -R$R -T -O >> $PS` 只写入文件尾，见 `-K` 和 `-O` 选项的介绍
3. `psxy -J$J -R$R -T -X10c -Y10c >> $PS` 只移动坐标原点而不绘制任何图形

`-W[-|+][<pen>][<attr>]` 设置线段或符号轮廓的画笔属性。

1. `<pen>` 见 [pen](#) 一节

2. -W+ 表示通过 -C 选项的 CPT 文件同时查找填充色和轮廓色
3. -W- 表示通过 -C 选项的 CPT 文件查找轮廓的颜色并关闭符号的填充
4. -W 选项后还可以加上额外的选项，可以指定线条的额外属性，见 [lines](#) 一节

## -S 选项

使用 -S 选项，则表示要绘制符号。-S 选项的基本语法是：

```
-S[<symbol>][<size>[<u>]]
```

其中 <symbol> 指定了符号类型，<size> 为符号的大小，<u> 为 <size> 的单位。

不同的符号类型，需要的输入数据格式也不同，但可以统一写成（用 ... 代表某符号特有的输入列）：

```
X    Y    ...
```

-S-|+|a|c|d|g|h|i|n|s|t|x|y|p 绘制一些简单的符号。

这几个符号比较简单，输入数据中不需要额外的列：

- -S- ：短横线，<size> 是短横线的长度；
- -S+ ：加号，<size> 是加号的外接圆的直径；
- -Sa ：星号 (star)，<size> 是星号的外接圆直径；
- -Sc ：圆 (circle)，<size> 为圆的直径；
- -Sd ：钻石 (diamond)，<size> 为外接圆直径；
- -Sg ：八角形 (octagon)，<size> 为外接圆直径；
- -Sh ：六边形 (hexagon)，<size> 为外接圆直径；
- -Si ：倒三角 (inverted triangle)，<size> 为外接圆直径；
- -Sn ：五角形 (pentagon)，<size> 为外接圆直径；
- -Sp ：点，不需要指定 <size> ，点的大小始终为一个像素点；
- -Ss ：正方形 (square)，<size> 为外接圆直径；
- -St ：三角形 (triangle)，<size> 为外接圆直径；
- -Sx ：叉号 (cross)，<size> 为外接圆直径；
- -Sy ：短竖线，<size> 为短竖线的长度；

对于小写符号 acdghinst ，<size> 表示外接圆直径；对于大写符号 ACDGHINST ，<size> 表示符号的面积与直径为 <size> 的圆的面积相同。

下图给出了上面所给出的 symbol 所对应的符号：

—	+	☆	○	◇	⬡	⬢	▽	⬠	□	△	×	
<b>-S-</b>	<b>-S+</b>	<b>-Sa</b>	<b>-Sc</b>	<b>-Sd</b>	<b>-Sg</b>	<b>-Sh</b>	<b>-Si</b>	<b>-Sn</b>	<b>-Ss</b>	<b>-St</b>	<b>-Sx</b>	<b>-Sy</b>
1	2	3	4	5	6	7	8	9	10	11	12	13

图 102.2: psxy -S 选项示意图

除了上述简单的符号之外，还有更多复杂的符号。

**-Sb|B[[<size><u>]][b[<base>]]** 绘制垂直 bar。

**-Sb** 用于在 X 坐标处绘制一个从 <base> 到 Y 位置的垂直 bar。

1. <size> 是 bar 宽度，其单位可以是长度单位 `clilp`，也可以用 `u` 表示 X 方向单位
2. 若不指定 `b<base>`，其默认值为 `ymin`
3. 指定 `b<base>`，为所有数据点指定 `base` 值
4. 加上 `b` 但未指定 `<base>`，则需要额外的一列数据来指定 `base` 的值
5. **-SB** 与 **-Sb** 类似，区别在于 **-SB** 绘制水平 bar

```
gmt psxy -R0/10/0/5 -JX15c/5c -B1 -Sb1cb > test.ps << EOF
2 3 1 0.5
4 2 1 1.5
8 4 1 2.5
EOF
```

**-Se|E** 绘制椭圆

**-Se** 用于绘制椭圆。对于椭圆而言，<size> 是不需要的。此时输入数据的格式为：

X	Y	方向	长轴长度	短轴长度
---	---	----	------	------

其中方向是相对于水平方向逆时针旋转的角度，两个轴的长度都使用长度单位，即 `clilp`

**-SE** 选项与 **-Se** 类似，区别在于：

- 第三列为方位角（相对于正北方向旋转的角度）。该角度会根据所选取的地图投影变换成角度
- 对于线性投影，长短轴的长度单位为数据单位，即与 **-R** 中数据范围的单位相同
- 对于地理投影，长轴和短轴的长度单位为千米，且不可更改

用长度单位指定一个椭圆：

```
echo 180 0 45 5c 3c | gmt psxy -R0/360/-90/90 -JN15c -B60 -Se > test.ps
```

线性投影下 **-SE** 的长短轴的单位为数据单位：

```
echo 180 0 45 300 100 | gmt psxy -R0/360/-90/90 -JX10c -B60 -SE > test.ps
```

地理投影下 -SE 的长短轴的单位是千米:

```
echo 80 0 45 22200 11100 | gmt psxy -R0/360/-90/90 -JN15c -B60 -SE > test.ps
```

若长短轴长度相等，则椭圆退化成圆，可以用于绘制直径以千米为单位的圆，从而解决了 -Sc 只能用长度单位而不能用距离单位画圆的不足。这一特性可以用于绘制等震中距线。比如如下命令可以绘制 30 度等震中距线:

```
echo 80 0 0 6660 6660 | gmt psxy -R0/360/-90/90 -JN15c -B60 -SE > test.ps
```

上面示例的输入数据中，方向和短轴长度都是多余的，所以 GMT 提供了 -SE- [<size>] 选项用于绘制直径为 <size> 的圆，若未指定 <size>，则需要在数据中指定圆的直径。比如 30 度和 60 度等震中距线可以用如下命令绘制:

```
gmt psxy -R0/360/-90/90 -JN15c -B60 -SE- > test.ps << EOF
180 0 6660
180 0 13320
EOF
```

-Sf<gap>[/<size>] [+l|+r] [+b+c+f+s+t] [+o<offset>] [+p<pen>]] 绘制 front，即在线段上加上符号以表示断层等 front

1. <gap> 线段上符号之间的距离，若 <gap> 为负值，则解释为线段上符号的个数
2. <size> 为符号大小
  - (a) 若省略了 <size>，则默认为 <gap> 的 30%
  - (b) 若 <gap> 为负值，则 <size> 是必须的
3. +l 和 +r 分别表示将符号画在线段的左侧还是右侧，默认是绘制在线段中间
4. +b 符号为 box
5. +c 符号为 circle
6. +t 符号为 triangle
7. +f 符号表示断层 (fault)，默认值。
8. +s 符号表示断层的滑动 (slip)，用于表示左旋或右旋断层。其可以接受一个可选的参数来控制绘制矢量时的角度
9. +o<offset> 将线段上的第一个符号相对于线段的起点偏离 <offset> 距离，默认值为 0
10. 默认符号的颜色与线段颜色相同 (-W 选项)，可以使用 +p<pen> 为符号单独指定颜色，也可以使用 +p，即不绘制符号的轮廓。

下面的例子分别绘制了 +b、+c、+f、+s、+t 所对应的符号:

```
#!/bin/bash
R=150/200/20/50
J=M15c
PS=psxy_-Sf.ps
gmt psbasemap -R$R -J$J -B10 -K > $PS
gmt psxy -R$R -J$J -Sf2c/0.1i+l+b -Gblack -W -K -0 >> $PS << EOF
155 30
160 40
EOF
gmt psxy -R$R -J$J -Sf2c/0.1i+l+c -Gblue -W -K -0 >> $PS << EOF
165 30
170 40
EOF
gmt psxy -R$R -J$J -Sf2c/0.1i+l+f -Gred -W -K -0 >> $PS << EOF
175 30
180 40
EOF
gmt psxy -R$R -J$J -Sf2c/0.3i+l+s+o1 -Gyellow -W -K -0 >> $PS << EOF
185 30
190 40
EOF
gmt psxy -R$R -J$J -Sf1c/0.1i+l+t -Gwhite -W -B10 -K -0 >> $PS << EOF
195 30
200 40
EOF
gmt psxy -R$R -J$J -T -0 >> $PS
rm gmt.*
```

## -Sj|J 绘制旋转矩形

其输入数据为:

X	Y	方向	X 轴长度	Y 轴长度
---	---	----	-------	-------

方向为相对于水平方向逆时针旋转的角度。

-SJ 与 -Sj 类似，区别在于：

1. 输入的第三列是方位角
2. 对于地理投影，X 轴和 Y 轴长度的单位为千米
3. 对于线性投影，X 轴和 Y 轴长度的单位与 -R 选项中数据范围的单位相同

**-Sk<name>/<size>** 绘制自定义的符号。

GMT 支持自定义符号，该选项会依次在当前目录、~/.gmt、\$GMT\_SHAREDIR/custom 目录中寻找自定义符号的定义文件 <name>.def。定义文件中的符号默认其大小为 1，然后会根据 <size> 对其进



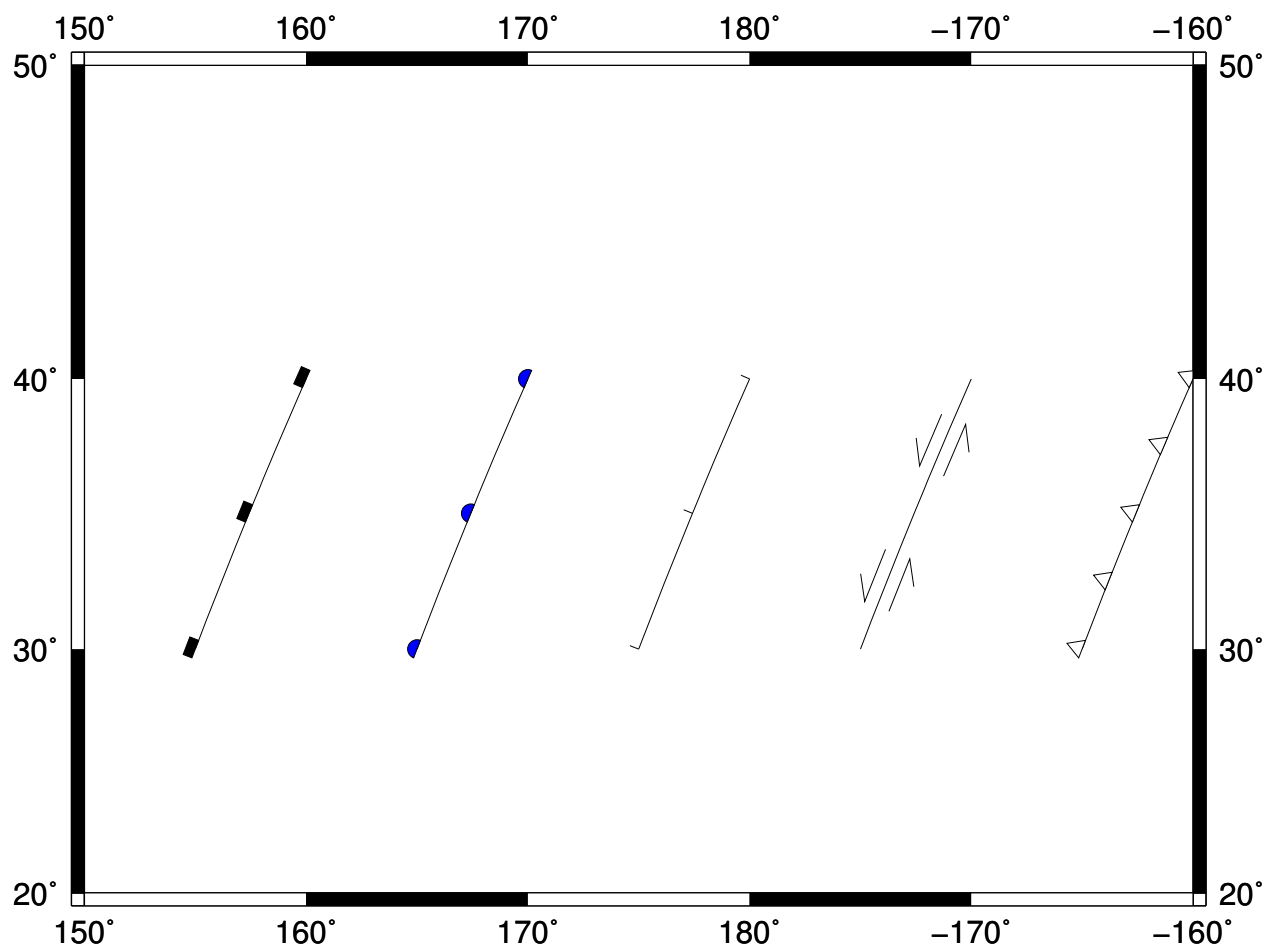


图 102.3: psxy -Sf 示意图

行缩放。关于如何自定义符号，请参考官方文档。

**-Sl<size>+t<string>+j<justify>** 绘制文本字符串

该选项的功能与 *ps\_text* 类似，不知道为何要设计这个选项。

1. <size> 文本串的大小
2. +t<string> 指定文本串
3. +j<justify> 修改文本串的对齐方式，默认为 CM

**-Sm|M<size>** 绘制数学圆弧

输入数据的格式为:

X	Y	radius_of_arc	start_direction	stop_direction
---	---	---------------	-----------------	----------------

1. <size> 为矢量箭头的长度
2. 圆弧的线宽由 -W 选项设定
3. -SM 选项与 -Sm 完全相同，只是当圆弧的夹角恰好是 90 度是，-SM 会用直角符号来表示
4. 圆弧的两端可加上额外的箭头，见 *vectors* 一节

```
#!/bin/bash
R=0/4/0/3
J=x2c
PS=psxy_-Sm.ps

gmt psbasemap -R$R -J$J -Ba1g1 -BWSen -K > $PS
gmt psxy -R$R -J$J -Sc0.15c -Gblack -K -0 >> $PS << EOF
1 1
3 1
EOF
gmt psxy -R$R -J$J -Sm0.2c+b+e+g -Gblack -W0.5p,red -K -0 >> $PS << EOF
1 1 1 10 60
EOF
gmt psxy -R$R -J$J -Sm0.2c+b+l -Gblack -W0.5p,blue -K -0 >> $PS << EOF
3 1 1 10 150
EOF
gmt psxy -R$R -J$J -T -0 >> $PS

rm gmt.*
```

**-Sq[<type>]<info>[:<labelinfo>]** 绘制 quoted lines，即带标注的线段，比如等值线、带断层名的断层线等

<type> 有 6 种可选的方式：

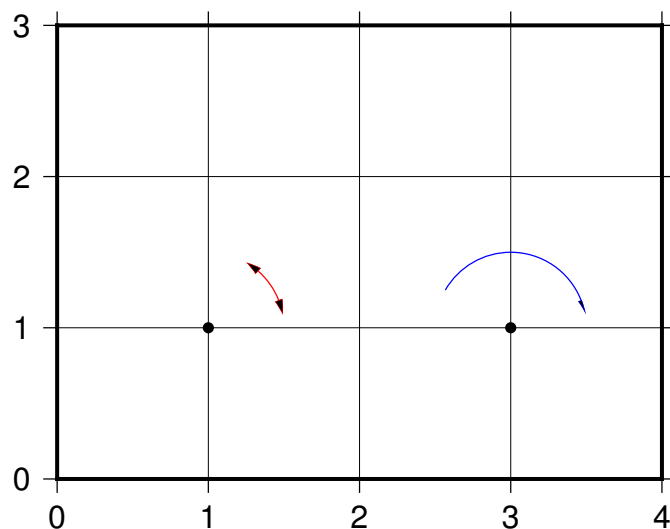


图 102.4: psxy -Sm 示意图

1. `d<dist>[<u>]/[<frac>]` 指定标签之间的距离，单位 `<u>` 为 `clip`；`<frac>` 表示将第一个标签放在距离 quoted lines 起点 `<frac>*<dist>` 处
2. `D<dist>[<u>]/[<frac>]` 指定标签之间的距离，单位 `<u>` 可以取 `e|f|k|M|n|u|d|m|s`
3. `f<ffile.d>` 根据 ASCII 文件 `<ffile.d>` 的内容确定标签的位置。仅当 `<ffile.d>` 中指定的标签位置与 quoted lines 上数据点的位置完全匹配时才会被绘制
4. `l<line1>[,<line2>,...]` 指定一个或多个以逗号分隔的线段，标签会放在线段与 quoted line 相交的地方。`<line>` 的格式为 `start_lon/start_lat/stop_lon/stop_lat`，其中 `start_lon/start_lat` 以及 `stop_lon/stop_lat` 可以用锚点中的两字符替换。
5. `L<line1>[,<line2>,...]` 与 `l` 类似，只是将线段解释为两点之间的大圆路径
6. `n<n_label>` 指定等间隔标签的数目，见官方文档
7. `N<n_label>` 见官方文档
8. `s<n_label>` 见官方文档
9. `S<n_label>` 见官方文档
10. `x<xfile.d>` 见官方文档
11. `X<xfile.d>` 见官方文档

`<labelinfo>` 用于控制标签的格式，其可以是下面子选项的任意组合，详情见官方文档：

1. `+a<angle>`
2. `+c<dx>/<dy>`
3. `+d`
4. `+e`

5. +f<font>
6. +g<color>
7. +j<just>
8. +l<label>
9. +L<label>
10. +n<dx>/<dy>
11. +o
12. +p<pen>
13. +r<min\_rad>
14. +t[<file>]
15. +u<unit>
16. +v
17. +w
18. +x[<first>,<last>]

#### -Sr 绘制矩形

“ 对该符号无效，其输入格式为：

X	Y	X 轴长度	Y 轴长度
---	---	-------	-------

#### -SR 绘制圆角矩形

<size> 对该符号无用。其输入格式为：

X	Y	X 轴长度	Y 轴长度	圆角半径
---	---	-------	-------	------

#### -Sv|V|= 绘制矢量

-Sv 用于绘制矢量，输入数据格式为：

X	Y	方向	长度
---	---	----	----

1. <size> 为矢量箭头的长度
2. 矢量宽度由 -W 控制
3. 更多箭头的属性见 [vectors](#) 一节
4. -SV 与 -Sv 类似，区别在于第三列是方位角而不是方向
5. -S= 与 -SV 类似，区别在于第四列长度的单位是千米

```
echo 2 2 45 5c | gmt psxy -R0/10/0/10 -JX10c/10c -B1 -Sv1c+e -W2p > test.ps
```

**-Sw|W** 绘制楔形饼图 (pie wedge)，即饼图中的一个切片

楔形饼图所需要的输入数据格式为:

X	Y	start_direction	stop_direcrion
---	---	-----------------	----------------

1. <size> 是楔形饼图所对应的圆的直径
2. 对于 -Sw ，第三、四列是楔形的开始和结束方向，其中方向定义为相对于 X 轴正方向（即东向）逆时针旋转的角度
3. 对于 -SW ，第三、四列是楔形的开始和结束方位角，其中方位角定义为相对于北向顺时针旋转的角度

下面的示例分别用 -SW 和 -Sw 画了两个不同大小的楔形饼图：

```
#!/bin/bash
R=0/10/0/5
J=x1c
PS=psxy_-Sw.ps
gmt psbasemap -R$R -J$J -Ba1g1 -K > $PS
gmt psxy -R$R -J$J -Sw2c -Glightblue -K -0 >> $PS << EOF
2 2 0 45
EOF

gmt psxy -R$R -J$J -SW3c -Glightblue -K -0 >> $PS << EOF
6 2 0 45
EOF
gmt psxy -R$R -J$J -T -0 >> $PS

rm gmt.*
```

**-S~[d|D|f|l|L|n|N|s|S|x|X]<info>[:<symbolinfo>]** 绘制 decorated line，即带有符号的线段。详见官方文档。

## 输入数据格式

-S 选项相对复杂，与不同的选项连用，或者后面接不同的参数，所需要的输入数据的格式也不同。不管是什么符号，至少都需要给定符号的位置，即 X 和 Y 是必须的：

X	Y
---	---

不同的符号，可能还需要额外的信息，统一写成（用 ... 代表某符号特有的输入列）：

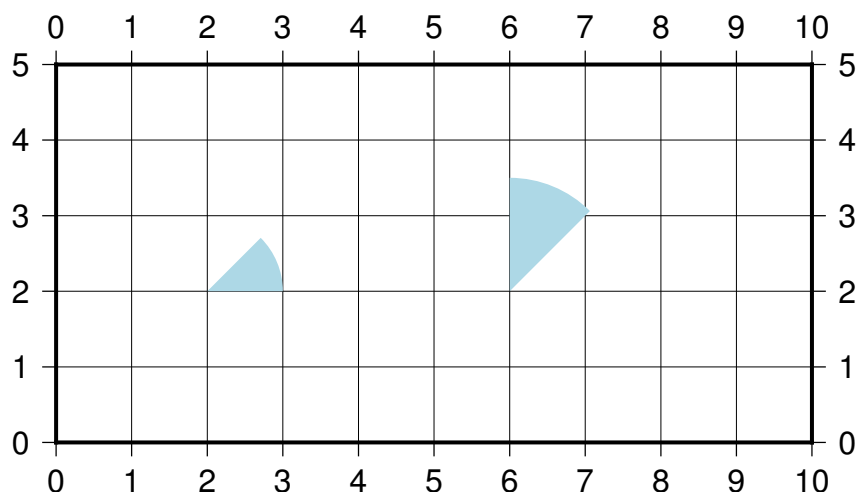


图 102.5: psxy -Sw 示意图。  
左边-Sw，右边-SW；图中 1 格表示 1cm。

```
X  Y  ...
```

若 -S 指定了符号类型但未指定大小，即 -S<symbol>，若该符号类型需要指定大小，则需要将符号大小放在输入数据的**第三列**，其他输入数据的列号延后，此时数据格式为：

```
X  Y  size  ...
```

若 size<=0，则跳过该记录行。

若 -S 选项后未指定符号代码，则符号代码必须位于输入文件的**最后一列**

```
X  Y  ...  symbol
```

若使用了 -C 和 -S 选项，则符号的填充色由数据的第三列决定，其他字段依次后移：

```
X  Y  [Z]  ...  symbol
```

因而总结一下输入数据的格式为：

```
x  y  [Z]  [size]  ...  [symbol]
```

其中 ... 为某些符号所要求的特殊的数据列，symbol 是未指定符号时必须的输入列，size 是未指定大小时的输入列。

## 多段数据

对于多段数据而言，每段数据的头段记录中都可以包含一些选项，以使得不同段数据拥有不同的属性。头段记录中的选项会覆盖命令中选项的参数：

- `-Gfill` : 设置当前段数据的填充色
- `-G-` : 对当前数据段关闭填充
- `-G` : 恢复到默认填充色
- `-W<pen>` : 设置当前段数据的画笔属性
- `-W` : 恢复到默认画笔属性 `MAP_DEFAULT_PEN`
- `-W-` : 不绘制轮廓
- `-Z<zval>` : 从 `cpt` 文件中查找 `Z` 值 `<zval>` 所对应的颜色作为填充色
- `-ZNaN` : 从 `cpt` 文件中获取 `NaN` 颜色

## 示例

最简单的命令，绘制线段或多边形，此时数据输入需要两列，即 `X` 和 `Y`:

```
gmt psxy -R0/10/0/10 -JX10c -B1 > test.ps << EOF
3 5
5 8
7 4
EOF
```

下面的脚本展示了 `-F` 选项的用法:

```
#!/bin/bash
ps=psxy_-F.ps

# 此处定义了函数 plotpts，用于绘制图中绿色和蓝色的圆圈，并加上文字
function plotpts
{
    # Plots the two data tables and places given text
    gmt psxy -R -J -O -K -Sc0.2c -Ggreen -Wfaint t1.txt
    gmt psxy -R -J -O -K -Sc0.2c -Gblue -Wfaint t2.txt
    echo $* | gmt pstext -R -J -O -K -F+cTL+jTL+f12p -Dj0.05i
}

# 生成测试用的第一个数据文件
cat << EOF > t1.txt
10      10
48      15
28      20
>
40      40
30      5
5       15
```

```

EOF
# 生成测试用的第二个数据文件
cat << EOF > t2.txt
7      20
29     11
8      4
EOF
scl=0.06i
# -Fcs: 默认的连接方式
gmt psxy -R0/50/0/45 -Jx${scl} -P -Ba10 -BWSne -Wlp t[12].txt -K > $ps
plotpts TWO DATA TABLES >> $ps

# -Fra
gmt psxy -R -J -Ba10 -BwSnE -Wlp t[12].txt -Fra -O -K -X3.25i >> $ps
plotpts DATASET ORIGIN >> $ps

# -FrF
gmt psxy -R -J -Wlp -Ba10 -BWSne t[12].txt -FrF -O -K -X3.25i -Y3.15i >> $ps
plotpts TABLE ORIGIN >> $ps

# -Frs
gmt psxy -R -J -Ba10 -BwSnE -Wlp t[12].txt -Frs -O -K -X3.25i >> $ps
plotpts SEGMENT ORIGIN >> $ps

# -Fr10/35
gmt psxy -R -J -Ba10 -BWSne -Wlp t[12].txt -Fr10/35 -O -K -X3.25i -Y3.15i >> $ps
plotpts FIXED ORIGIN >> $ps
echo 10 35 | gmt psxy -R -J -O -K -Sa0.4c -Gred -Wfaint >> $ps

# -Fna
gmt psxy -R -J -Ba10 -BwSnE -Wlp t[12].txt -Fna -O -K -X3.25i >> $ps
plotpts NETWORK >> $ps

gmt psxy -R -J -O -T >> $ps
rm gmt.* t[12].txt

```

-L 选项的示例:

```

#!/bin/bash
ps=psxy_-L.ps
cat << EOF > t.txt
1 1
2 3
3 2
4 4

```



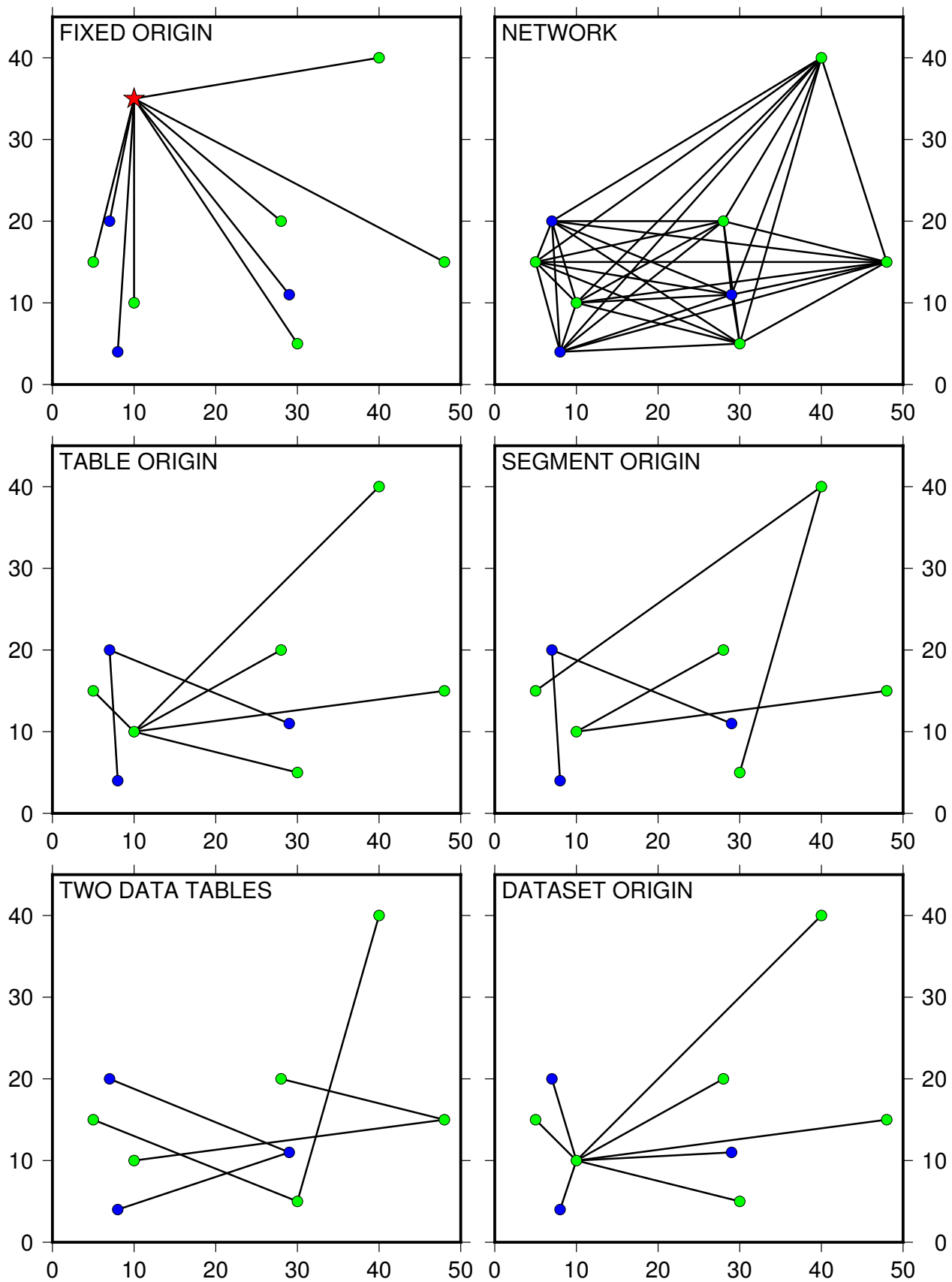


图 102.6: psxy -F 选项示意图

EOF

```
gmt psxy -R0/5/0/5 -JX3i -K -B0 t.txt -Gred -W2p -L+x1 > $ps
gmt psxy -R -J -0 -K -B0 t.txt -Gred -W2p -L+xr -X3.25i >> $ps
gmt psxy -R -J -0 -K -B0 t.txt -Gred -W2p -L+x4.5 -X3.25i >> $ps
gmt psxy -R -J -0 -K -B0 t.txt -Gred -W2p -L+yt -X-6.5i -Y3.25i >> $ps
gmt psxy -R -J -0 -K -B0 t.txt -Gred -W2p -L+yb -X3.25i >> $ps
gmt psxy -R -J -0 -K -B0 t.txt -Gred -W2p -L+y4 -X3.25i >> $ps
gmt psxy -R -J -0 -T >> $ps
rm gmt.* t.txt
```

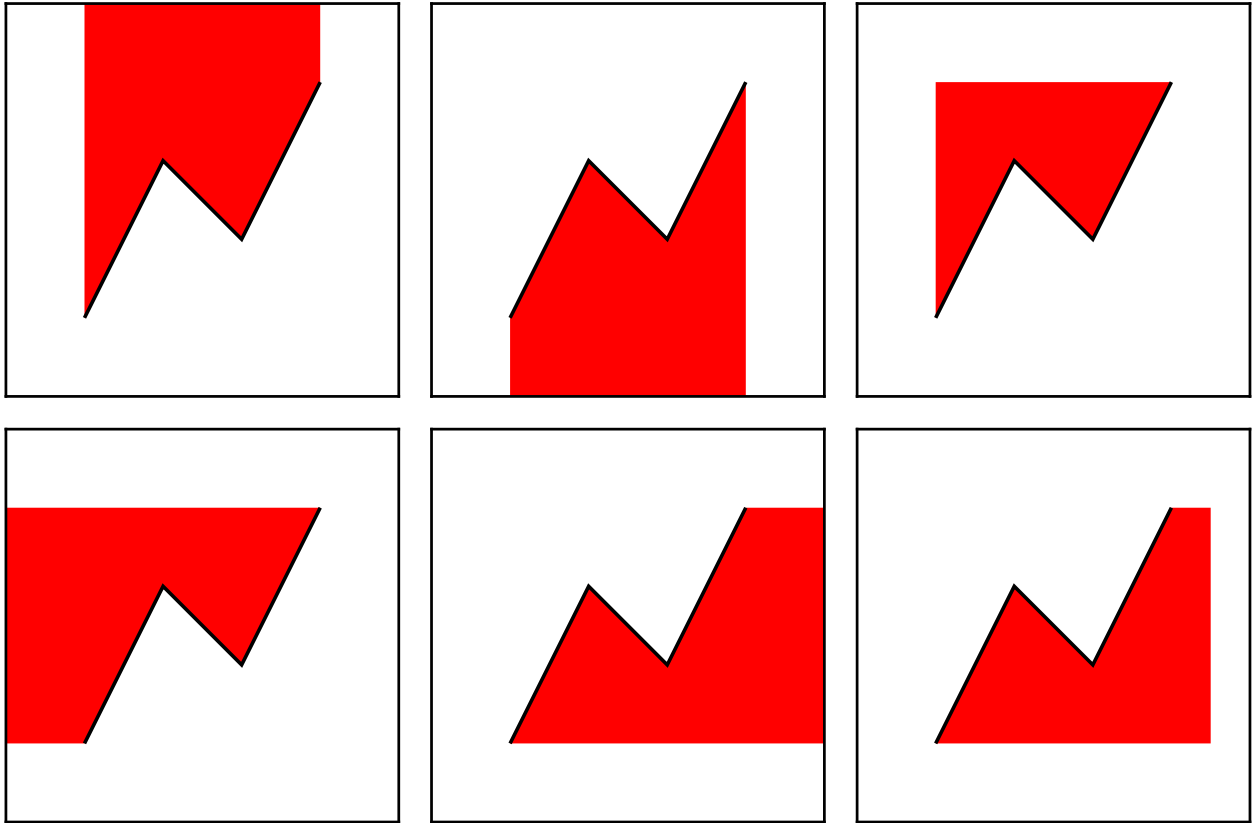


图 102.7: psxy -L 选项示意图

---

psxyz

---

官方文档 [psxyz](#)

---

**rotconverter**

---

官方文档 [rotconverter](#)

---

**sample1d**

---

官方文档 [sample1d](#)

**简介** 对 1D 表数据进行重采样

该命令既可以对常规的一维数据（比如时间序列，自变量为时间）进行重采样，也可以对地理坐标下的测线进行重采样。

## 选项

**<table>** 多列表数据，其中某一列是自变量，其他列为因变量。自变量所在列必须递增或递减。

**-Af|p|m|r|R[+l]** 指定插值方式。

- **-Af** 保留原始数据点，若有必要，则在原始数据点的中间加上额外的点
- **-Am** 对测线进行采样时，先沿着 Y 方向，再沿着 X 方向
- **-Ap** 对测线进行采样时，先沿着 X 方向，再沿着 Y 方向
- **-Ar** 等间距采样
- **-AR** 等间距采样，但会调整间距以适应自变量的原始长度
- **+l** 见官方文档

**-Fl|a|c|n[+1|+2]** 插值方式

- **l** 线性插值
- **a** Akima 样条插值
- **c** natural cubic spline
- **n** 不插值，取最近的数据点作为插值后的值
- **+1|+2** 插值的同时计算 spline 的一阶或二阶插值

**-I<inc>[<unit>]** 默认的等间隔采样间隔是自变量第一个和第二个数据点的间隔，该选项可以自定义采样间隔 **<inc>** 。

加上 **<unit>** 表明数据文件的前两列包含经纬度信息，重采样后的测线的采样间隔的单位是 **<units>**。若想要采样笛卡尔坐标下的 (x,y)，则需要指定单位为 **c** 。

**-N<knotfile>** **<knotfile>** 中包含了一系列 X 坐标轴，使用该选项则会将原始数据插值到这些 X 坐标轴数据点上。

**-S<start>[/<stop>]** 对于等间隔采样而言，**<start>** 是第一个输出值的 X 位置，**<stop>** 是最后一个输出值的 X 位置。

**-T<col>** 指定输入数据中的哪列数据是自变量。

## 示例

输入数据的格式为:

```
time distance gravity magnetics bathymetry
```

使用 Akima spline 插值方式将其采样为 1 千米等间隔:

```
gmt sample1d profiles.tdgmb -I1 -Fa -T1 > profiles_equi_d.tdgmb
```

将 0 到 6 之间的数据用 cubic spline 方式重采样为 0.01 间隔，不输出数据而是输出一阶偏导（即斜率）:

```
gmt sample1d points.txt -S0/6 -I0.01 -Fc+1 > slopes.txt
```

测线数据中包含经度、纬度和深度，将其采样为每 2 海里一个点:

```
gmt sample1d track.txt -I2n -AR > new_track.dt
```

同上，但确保包含了原始数据点:

```
gmt sample1d track.txt -I2n -Af > new_track.dt
```

To obtain a rhumb line (loxodrome) sampled every 5 km instead:

```
gmt sample1d track.txt -I5k -AR+1 > new_track.dt
```

---

**seggy2grd**

---

官方文档 [seggy2grd](#)

---

**spectrum1d**

---

官方文档 [spectrum1d](#)



---

**sph2grd**

---

官方文档 [sph2grd](#)

---

**sphdistance**

---

官方文档 [sphdistance](#)

---

**sphinterpolate**

---

官方文档 [sphinterpolate](#)

---

**sphtriangulate**

---

官方文档 [sphtriangulate](#)

---

**splitxyz**

---

官方文档 [splitxyz](#)

---

**surface**

---

官方文档 [surface](#)

---

**talwani2d**

---

官方文档 [talwani2d](#)

---

**talwani3d**

---

官方文档 [talwani3d](#)



---

**trend1d**

---

官方文档 [trend1d](#)

---

**trend2d**

---

官方文档 [trend2d](#)

---

**triangulate**

---

官方文档 [triangulate](#)

---

**x2sys\_binlist**

---

官方文档 [x2sys\\_binlist](#)

---

**x2sys\_cross**

---

官方文档 [x2sys\\_cross](#)

---

**x2sys\_datalist**

---

官方文档 [x2sys\\_datalist](#)

---

**x2sys\_get**

---

官方文档 [x2sys\\_get](#)

---

**x2sys\_init**

---

官方文档 [x2sys\\_init](#)



---

**x2sys\_list**

---

官方文档 [x2sys\\_list](#)

---

**x2sys\_merge**

---

官方文档 [x2sys\\_merge](#)

---

**x2sys\_put**

---

官方文档 [x2sys\\_put](#)

---

x2sys\_report

---

官方文档 [x2sys\\_report](#)

---

**x2sys\_solve**

---

官方文档 [x2sys\\_solve](#)

## xyz2grd

官方文档 [xyz2grd](#)

**简介** 将 XYZ 数据或 Z 数据转换成网格文件

该命令读取一个或多个 Z 数据或 XYZ 数据，并将其转换成二进制网格文件。

## 选项

**<table>** 输入表数据。可以只包含 Z 值，也可以包含 (x,y,z) 值。可以是 ASCII 格式，也可以是二进制格式。XYZ 数据不要求排序，Z 数据则必须排序，且必须使用 **-Z** 选项。

**-G<grdfile>** 生成的网格文件名

**-I<xinc>[<unit>][=|+]/<yinc>[<unit>][=|+]** 指定 X 和 Y 方向的网格间隔

- **<xinc>** X 方向的网格间隔
- **<yinc>** Y 方向的网格间隔
- **<unit>** 网格间隔的单位
- **=** 微调 X 和 Y 方向范围的最大值，使得其是网格间隔的整数倍（默认会微调网格间隔以适应给定的数据范围）
- **+** 表明 **<xinc>** 和 **<yinc>** 不是网格间隔，而是 X 和 Y 方向的节点数

**-A[f|l|m|n|r|s|u|z]** 多个数据落在同一个网格节点内时的处理方式。

默认情况下，若有多个数据落在同一个网格节点内，会将这些数据的均值作为该节点的值。使用该选项可以控制这一行为：

- **-Af** 将第一个落在该节点内的数据值作为节点值
- **-As** 将最后一个落在该节点内的数据值作为节点值
- **-Al** 将落在该节点内的所有值的最小值作为节点值

- **-Au** 将落在该节点内的所有值的最大值作为节点值
- **-Am** 将落在该节点内的所有值的均值作为节点值
- **-Ar** 将落在该节点内的所有值的 RMS 值作为节点值
- **-An** 将落在该节点内的数据数目作为该节点值，该选项仅要求输入数据中有 XY 值
- **-Az** 将落在该节点内的所有值的和作为节点值

**-D<xname>/<yname>/<zname>/<scale>/<offset>/<invalid>/<title>/<remark>** 设置网格文件中的一些辅助信息

- **<xname>/<yname>/<zname>** 变量名及其单位，比如 "distance [km]"
- **<scale>** 读入网格后要乘以的缩放因子（默认值为 1）
- **<offset>** 乘以缩放因子后要加上的偏移量（默认值为 0）
- **<invalid>** 用于表示数据缺失的值（默认值为 NaN）
- **<title>/<remark>** 任意信息

若需要保持某些值不变，可以直接将该字段留空。若想要在某个字段中使用字符 / ，则必须使用其他字符（不能是字母、数字、等号）作为分隔符，比如：

```
-D:<xname>:<yname>:<zname>:<scale>:<offset>:<invalid>:<title>:<remark>:
```

**-S[<zfile>]** 对输入的 Z 文件做字节序转换并输出到标准输出或保存到文件 <zfile> 中。该选项不会只做字节序转换，不生成网格文件。该选项必须与 **-Z** 选项连用。

**-Z[<flags>]** 指定 Z 数据的格式。见官方文档

backtracker, [11](#)  
blockmean, [12](#)  
blockmedian, [13](#)  
blockmode, [14](#)  
  
dimfilter, [15](#)  
  
filter1d, [16](#)  
fitcircle, [18](#)  
  
gmt, [20](#)  
gmt2kml, [22](#)  
gmt5syntax, [23](#)  
gmtconnect, [24](#)  
gmtconvert, [26](#)  
gmtdefaults, [27](#)  
gmtflexure, [28](#)  
gmtget, [29](#)  
gmtgravmag3d, [30](#)  
gmtinfo, [31](#)  
gmtlogo, [33](#)  
gmtmath, [34](#)  
gmtregress, [35](#)  
gmtselect, [36](#)  
gmtset, [39](#)  
gmtsimplify, [41](#)  
gmtspatial, [42](#)  
gmtswitch, [43](#)  
gmtvector, [44](#)  
gmtwhich, [45](#)  
gravfft, [46](#)  
grd2cpt, [47](#)  
  
grd2rgb, [48](#)  
grd2xyz, [50](#)  
grdblend, [52](#)  
grdclip, [54](#)  
grdcontour, [56](#)  
grdconvert, [84](#)  
grdcut, [57](#)  
grdedit, [59](#)  
grdfft, [61](#)  
grdfilter, [62](#)  
grdflexure, [63](#)  
grdgradient, [64](#)  
grdgravmag3d, [65](#)  
grdhisteq, [66](#)  
grdimage, [67](#)  
grdinfo, [69](#)  
grdlandmask, [72](#)  
grdmask, [74](#)  
grdmath, [76](#)  
grdpaste, [77](#)  
grdpmodeler, [78](#)  
grdproject, [79](#)  
grdraster, [81](#)  
grdredpol, [83](#)  
grdrotater, [86](#)  
grdsample, [87](#)  
grdseamount, [89](#)  
grdspotter, [90](#)  
grdtrack, [91](#)  
grdtrend, [92](#)  
grdvector, [94](#)



grdview, [96](#)  
grdvolume, [97](#)  
greenspline, [99](#)  
gshhg, [100](#)  
  
hotspotter, [101](#)  
  
img2google, [102](#)  
img2grd, [103](#)  
isogmt, [104](#)  
  
kml2gmt, [106](#)  
  
makecpt, [108](#)  
mapproject, [109](#)  
mgd77convert, [111](#)  
mgd77info, [112](#)  
mgd77list, [113](#)  
mgd77magref, [114](#)  
mgd77manage, [115](#)  
mgd77path, [116](#)  
mgd77sniffer, [117](#)  
mgd77track, [118](#)  
  
nearneighbor, [119](#)  
  
originator, [120](#)  
  
project, [121](#)  
psbasemap, [123](#)  
psclip, [132](#)  
pscoast, [134](#)  
pscontour, [140](#)  
psconvert, [141](#)  
pscoupe, [146](#)  
pshistogram, [152](#)  
psimage, [155](#)  
pslegend, [157](#)  
psmask, [165](#)  
psmeca, [166](#)  
pspolar, [174](#)  
psrose, [178](#)  
psscale, [181](#)  
pssegy, [183](#)

pssegyz, [184](#)  
pstext, [185](#)  
psvelo, [191](#)  
pswiggle, [196](#)  
psxy, [198](#)  
psxyz, [217](#)  
  
rotconverter, [218](#)  
  
sample1d, [219](#)  
segy2grd, [221](#)  
spectrum1d, [222](#)  
sph2grd, [223](#)  
sphdistance, [224](#)  
sphinterpolate, [225](#)  
sphtriangulate, [226](#)  
splitxyz, [227](#)  
surface, [228](#)  
  
talwani2d, [229](#)  
talwani3d, [230](#)  
trend1d, [231](#)  
trend2d, [232](#)  
triangulate, [233](#)  
  
x2sys\_binlist, [234](#)  
x2sys\_cross, [235](#)  
x2sys\_datalist, [236](#)  
x2sys\_get, [237](#)  
x2sys\_init, [238](#)  
x2sys\_list, [239](#)  
x2sys\_merge, [240](#)  
x2sys\_put, [241](#)  
x2sys\_report, [242](#)  
x2sys\_solve, [243](#)  
xyz2grd, [244](#)